Q 系列精简模式 CPU

用户手册

（功能解说、编程基础篇）

MITSUBISHI

Q系列

Q系列

可编程控制器

MELSEC-Q

Q00JCPU
Q00CPU
Q01CPU

●安全上的注意事项●

（使用前请务必阅读）

使用本产品时，请充分阅读本手册以及本手册所介绍的相关手册，同时，在充分注意安全的前提下，正确操作。

本安全上的注意事项中，将安全注意事项划分为“危险”和“注意”2个等级。



危险

操作失误的情况下，可能引起灾难性后果，有造成死亡或重伤的可能性。



注意

操作失误的情况下，可能引发危险事态，有造成中等程度伤害或轻伤的可能性，以及可能仅发生物质损伤。

此外，即使是⚠注意中所记载的事项，根据情况不同，仍存在引起严重后果的可能性。

两者所记载的都是重要内容，请务必遵守。

本手册须妥加保管，以便必要时查阅；同时，请务必交到最终使用者的手上。

【设计上的注意事项】



危险

● 请在可编程控制器的外部设置安全电路，以保证外部电源发生异常和可编程控制器出现故障时，系统整体仍能安全运行。误输出、误动作均有引发事故的可能。

（1）请在可编程控制器的外部构建紧急停止电路、保护电路、正转/反转等相反动作的连锁电路、定位的上限/下限等防止机械损坏的连锁电路等电路。

（2）可编程控制器一旦检测出以下的异常状态，在（a）的情况下就会停止运算并切断全部输出；在（b）的情况下则会停止运算，并根据参数的设置保持或切断全部输出。

（a）电源模块的过流保护装置或过压保护装置发生作用时。

（b）可编程控制器的CPU的警戒定时器出错等自我诊断功能检测出异常时。

此外，发生可编程控制器的CPU无法检测出的输出控制部分的异常时，有时可能所有的输出都为“通”。此时，为了保证机械的动作对安全侧发生效力，请在可编程控制器的外部构建故障保险电路，或设置机构。关于故障保险电路的实例，请参照本手册的“装载和设置”。

（3）根据输出模块的继电器、晶体管等的故障的不同，输出有可能一直为“通”不变，或一直为“断”不变。对于可能引起重大事故的输出信号，请设置外部监视电路。

【设计上的注意事项】

⚠危险

- 在输出模块上，因额定负载以上的电流或负载短路等而发生过电流长时间连续流过的情况下，存在冒烟、起火的危险，请在外部设置保险丝等的安全电路。

- 电路要设计成在可编程控制器主体的电源开启后，外部供电电源才开启的电路。
  如果先开启外部供给电源，就存在因误输出、误动作而引起事故的危险。

- 关于数据通信发生通信异常时的各个站电的动作状态，请参照各数据通信的手册。
  存在因误输出、误动作而引起事故的危险。

- 在CPU模块上连接外部设备，或在智能功能模块上连接个人电脑等，对运行中的可编程控制器进行控制（更改数据）时，请在顺控程序上构建连锁电路，以保证系统整体始终安全运行。
  此外，对运行中的可编程控制器进行其他控制（程序的更改、运行状态的更改（状态控制））时，请熟读手册，在充分确认安全的前提下再进行。
  特别是从外部设备对远地的可编程控制器进行上述控制时，有时可能会因为数据通信异常而无法即时地对可编程控制器方面的故障采取应对措施。
  在顺控程序上构建连锁电路的同时，请在外部设备和可编程控制器的CPU之间采取数据通信发生异常时的系统处置方法。

⚠注意

- 控制线和通信电缆请不要和主电路或动力电缆等捆扎在一起，也不要相互靠近。
  作为参考基准，请离开100 mm以上的距离。
  因为噪声可能成为误动作的原因。

- 在输出模块上对灯泡负载、加热器、螺线管阀门等进行控制时，输出从“断”→“通”的过程中有时会有大电流（达正常情况的10倍左右）流过，因此，请采取选用额定电流有充分余量的输出模块等的对策。

【安装上的注意事项】



- 可编程控制器请在《基本型QCPU（Q模式）用户手册》（硬件设计・保养检查篇）中规定的一般技术规范的环境下使用。
  如果在一般技术规范的范围以外使用，可能引发触电、火灾、误动作、产品损伤或性能变差。

- 边按下模块下部的模块安装手柄，边将模块固定用突起部可靠地插入并安装到基板的固定孔内。如果模块的安装不正确，会导致误动作、故障、跌落。
  在震动频繁的环境下使用时，请用螺丝对模块进行固定。
  螺丝的拧紧请在规定的扭矩范围内进行。
  如果螺丝拧得过松，就有可能引起脱落、短路或误动作。
  而如果螺丝拧得过紧，可能会导致螺丝或模块的破损从而引起跌落、短路或误动作。

- 扩展电缆请可靠地安装在基板的模块接插件上。
  安装后请进行检查。
  接触不良会引起误输入或误输出。

- 模块的拆装请务必在外部全相切断电源后再进行。
  不全相切断电源时，存在损伤产品的危险。

- 请不要直接接触模块的导电部分，
  否则引发模块误动作、故障。

【接线时的注意事项】



- 接线作业等请务必在外部全相切断电源后再进行。
  不全相切断电源时，存在触电或损伤产品的危险。

- 接线作业后进行通电、运行时，请务必安装附属于产品的端子盖。
  不安装端子盖时，存在触电的危险。

【接线上的注意事项】



- FG端子和LG端子为可编程控制器专用的D种接地（第三种接地）以上的接地，请务必接地。
  否则有触电、误动作的危险。

- 模块的接线请在确认产品的额定电压和端子排列的后正确地进行。
  如果连接了与额定电压不同的电源，或接线出错，就有可能成为火灾、故障的原因。

- 用于外部连接的接插件请用制造商指定的工具压装、压接或正确地焊接。
  接触不良，可能引起短路、火灾或误动作。

- 端子螺丝的紧固请在规定的扭矩范围内进行。
  端子螺丝拧得过松时，可能引起短路、火灾或误动作。
  而若端子螺丝拧得过紧，可能会导致螺丝或模块的破损从而引起跌落、短路或误动作。

- 模块内部请注意不要让切屑或接线时的碎屑等异物进入。
  否则会引起火灾、故障或误动作。

- 模块上部贴有防止混入的标贴，以防止接线时模块内部混入接线碎屑等异物。
  接线作业过程中，请不要剥下此标签。
  系统运行时，请务必将此标签剥下，以便散热。

【启动・保养时的注意事项】



- 通电时请不要接触端子。
  可能引起触电。

- 电池请正确连接。请不要对电池进行充电、分解、加热、投入火中、短路、焊锡等处理。
  出错地使用电池可能会因为发热、破裂、起火等引起人员受伤或火灾。

- 进行清扫或对端子螺丝、模块安装螺丝加固时，请务必在外部全相切断电源。
  不全相切断电源时，存在触电的危险。
  端子螺丝松动时，可能引起短路或误动作。
  而若螺丝拧得过紧，又有可能会因为螺丝或模块的破损而引起跌落、短路或误动作。

【启动・保养时的注意事项】



● 在运行中的CPU模块上连接外部设备后进行的在线操作（特别是程序的更改、强制输出、运行状态的更改）应在熟读本手册，并充分确认安全的基础上再进行。
否则可能因操作失误而引起机械的破损或事故。

● 请不要进行各模块的分解或改造。
否则可能会引起故障、误动作、人员受伤或火灾。

● 请离开可编程控制器25 cm以上使用移动电话或PHS等无线通信设备。
否则可能会引起误动作。

● 模块的拆装请务必在外部全相切断电源后再进行。
不全相切断电源，可能引起模块故障或误动作。

【报废时的注意事项】



● 产品报废时，请作为工业废品处理。

修订记录

※使用说明书的编号记在本说明书封底的左下角处。

印刷日期	※使用说明书编号	改　定　内　容
2002年7月	SH(NA)-080331C-A	第1次印刷

序　　言

此次承蒙购买MELSEC—Q系列可编程控制器，我们表示十分感谢。
使用前请仔细阅读本书，在充分理解MELSEC—Q系列可编程控制器的功能・性能的基础上，正确地使用。

目录

 A － 9

（相关手册）・・・・基本型QCPU（Q模块）用户手册

（硬件设计・保养检查篇）

目录

附录

关于手册

与本产品相关的手册包括以下几种。
请根据需要并参考本表订购。

相关手册

手册名称	手册编号 （条形码）
基本型QCPU（Q模块）用户手册 （硬件设计・保养检查篇） 阐述CPU模块、电源模块、基板、扩展电缆及存储卡等的技术规范。 （另购）	SH-080333C
QCPU（Q模块）/QnACPU 编程手册（通用指令篇） 阐述顺控指令、基本指令、应用指令及微机程序的使用方法。 （另购）	SH-080282C

本手册的使用方法

本手册是您使用基本型QCPU（Q00J/Q00/Q01CPU）时，为便于您理解必要的CPU模块的功能、程序、软元件等有关内容而提供的手册。

本手册的构成可大致划分如下。

① 第1章，第2章	阐述CPU模块的概要及系统构成等有关内容。 可借此了解CPU模块的特长、系统构建的基础知识。
② 第3章～第6章	阐述CPU模块的性能要求、可执行程序、输入输出编号、存储器等有关内容。
③ 第7章	阐述CPU模块的功能。
④ 第8章	阐述与智能功能模块之间的通信方法。
⑤ 第9章～第10章	阐述CPU模块的参数和软元件。
⑥ 第11章	阐述CPU模块的处理时间。
⑦ 第12章	阐述GX Developer向CPU模块写入参数、程序的步骤。

备　注

本手册中没有对电源模块、基板、扩展电缆、电池的技术规范等进行说明。
请参照下列手册。
• 基本型QCPU（Q模块）用户手册
（硬件设置 • 保养检查篇）

本手册中使用的总称和简称

除非特别注明，本手册中采用如下所示的总称和简称用以阐述基本型QCPU。

总称/简称	总称・简称的内容
基本型QCPU， CPU模块	Q00JCPU、Q00CPU、Q01CPU的总称。
Q00/Q01CPU	Q00CPU、Q01CPU的简称。
Q00JCPU/Q00/Q01CPU	Q00JCPU、Q00CPU、Q01CPU的简称
高性能模板QCPU	Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU的总称。
Q系列	三菱通用可编程控制器MELSEC—Q系列的简称。
GX Developer	GX Developer第7版及以后产品的简称。
主基板	可以安装Q00/Q01CPU、Q系列电源模块、输入输出模块、智能功能模块的Q33B、Q35B、Q38B、Q312B型主基板和Q00JCPU的基板的总称。
Q3□B	可以安装Q00/Q01CPU、Q系列电源模块、输入输出模块、智能功能模块的Q33B、Q35B、Q38B、Q312B型主基板的总称。
扩展基板	Q5□B、Q6□B的总称。
Q5□B	可以安装Q系列输入输出模块、智能功能模块的Q52B、Q55B型扩展基板的总称。
Q6□B	可以安装Q系列输入输出模块、输入输出模块、智能功能模块的Q63B、Q65B、Q68B、Q612B型扩展基板的总称。
基板	主基板、扩展基板的总称。
扩展电缆	QC05B、QC06B、QC12B、QC30B、QC50B、QC100B型扩展电缆的总称。
电源模块	Q61P—A1、Q61P—A2、Q62P、Q63P型电源模块和Q00JCPU电源部分的总称。
电池	Q6BAT型CPU模块用电池。

第1章 概要

本手册阐述基本型QCPU（Q00JCPU、Q00CPU、Q01CPU）的内置存储器、功能、程序、软元件等有关内容。

电源模块、基板、扩展电缆、电池的技术规范等请参照下列手册。
基本型QCPU（Q模块）用户手册

（硬件设计・保养检查篇）

（1）Q00JCPU
- Q00JCPU是电源模块、主基板（5个插槽）一体化的CPU模块。
- 扩展基板最多可连接2级，最多可以安装16块输入输出模块、智能功能模块等。
- 主基板、扩展基板上可以控制的输入输出点数为256点。

（2）Q00CPU、Q01CPU
- Q00CPU、Q01CPU是单独的CPU模块，安装在主基板上。
- 扩展基板最多可连接4级，最多可以安装24块输入输出模块、智能功能模块等。
- 主基板、扩展基板上可以控制的输入输出点数为1024点。

基本型QCPU（Q00JCPU、Q00CPU、Q01CPU）的不同点如下表所示。

项 目		Q00JCPU	Q00CPU	Q01CPU
CPU模块		CPU模块、电源模块、主基板（5个插槽）一体型	单体的CPU模块	
主基板		不要	要（Q33B、Q35B、Q38B、Q312B）	
扩展基板		可以连接（Q52B、Q55B、Q63B、Q65B、Q68B、Q612B）		
扩展级数		最多2级	最多4级	
可以安装的模块数		16块	24块	
电源模块				
主基板		不要	要（Q61P—A1、Q61P—A2、Q62P、Q63P）	
扩展基板模块	Q52B、Q55B	不要		
扩展基板模块	Q63B、Q65B、Q68B、Q612B	要（Q61P-A1、Q61P-A2、Q62P、Q63P）		
扩展电缆		QC05B、QC06B、QC12B、QC30B、QC50B、QC100B		
存储卡接口		无		
外部接口	RS-232	有（传输速率：9.6 kbps、19.2 kbps、38.4 kbps、57.6 kbps、115.2 kbps）		
外部接口	USB	无		
处理速度（顺控程序）	LD X0	0.20 μs	0.16 μs	0.10 μs
处理速度（顺控程序）	MOV D0 D1	0.70 μs	0.56 μs	0.35 μs
程序容量 ＊		8 k步（32 k字节）	8 k步（32 k字节）	14 k步（56 k字节）
存储容量	程序存储器	58 k字节	94 k字节	
存储容量	标准RAM	————	64 k字节	
存储容量	标准ROM	58 k字节	94 k字节	
软元件寄存器容量		18 k字节（可以在16.4 k字节范围内变更元件点数）		
输入输出软元件点数（包括远程输入输出）		2048点		
输入输出点数		256点	1024点	
文件寄存器		无	有（固定为32 k点）	
串行通信功能		无	有（使用CPU模块的RS—232接口）	

＊：程序容量的1步为4个字节。

1.1 特点

（1）可以控制多点输入输出

作为可以向基板上安装的输入输出模块存取的输入输出点数，Q00JCPU支持256点（X/Y0～FF）；Q00CPU/Q01CPU支持1024点（X/Y0～3FF）。

作为CC—Link的远程输入输出、MELSECNET/H的通信输入输出（LX、LY）的刷新可以使用的输入输出软元件点数，最多支持2048点（X/Y0～7FF）。

（2）根据程序容量的调节

可以选择与所使用的程序容量最相宜的CPU模块。

Q00JCPU、Q00CPU ： 8k步

Q01CPU ：14k步

（3）实现高速处理

LD指令（LD X0）的处理速度为如下所示的数值。

Q00JCPU ： 0.20μs

Q00CPU ： 0.16μs

Q01CPU ： 0.10μs

此外，利用MELSEC—Q系列用的基板的高速系统总线，可以实现对智能功能模块存取和网络通信刷新的高速化。

MELSECNET/H通信刷新处理 ：2.2 ms/2 k字 *1

*1： Q01CPU上不使用SB/SW，且MELSECNET/H网络模块安装在基板上的场合。

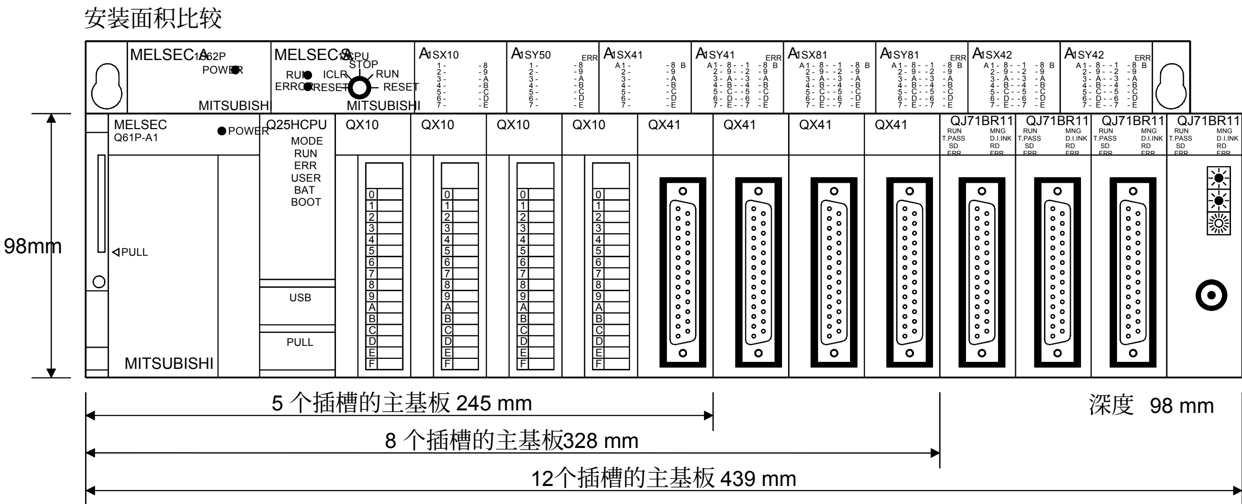
（4）通过和GX开发器的高速通信提高调试效率

基本型QCPU的RS—232接口可以以最大115.2 kbps的高速率进行程序的写入/读出或监视。

（5）通过小型化节省空间尺寸

基本型QCPU的安装面 和AnS系列相比60%。

安装面积比较



（6）最多可以连接4级/2级扩展基板

（a）Q00JCPU最多可以连接2级扩展基板（包括主基板为3级），最多可安装16个模块。

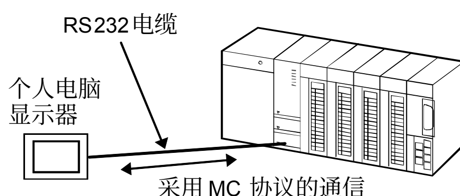
（b）Q00/Q01CPU最多可以连接4级扩展基板（包括主基板为5级），最多可安装24个模块。

（c）扩展电缆的合计延长距离最长为13.2 m，可以高度灵活地配置扩展基板。

要　点
GOT连接到总线上的情况下，由于GOT使用的是上述级数中的1级，因此，扩展基板的连接级数将减少1级。

（7）利用串行通信功能和个人电脑、显示器进行通信

可以将Q00CPU、Q01CPU的RS—232接口和个人电脑、显示器等连接，采用MELSEC通信协议（以下简称“MC协议”）实现通信。



串行通信功能可以使用的只有采用MC协议（QnA互换3C帧（格式4）、QnA互换4C帧（格式4、5））的通信。

串行通信功能不能进行无顺序模式、双工模式的通信。

关于MC协议，请参照下列手册。

• Q对应MELSEC通信协议参考手册

（8）内置标准ROM

通过将闪存ROM作为标准配置，用以存储ROM运行所使用的参数和顺控程序，重要的程序保护变得容易了。

（9）CC—Link系统的操作简单

使用1块CC—Link系统的主控模块时，可以在没有参数的情况下，最多对32个远程输入/输出站的输入输出信号进行控制。

此外，可以象控制基板上的输入输出模块一样对远程输入输出站进行控制。

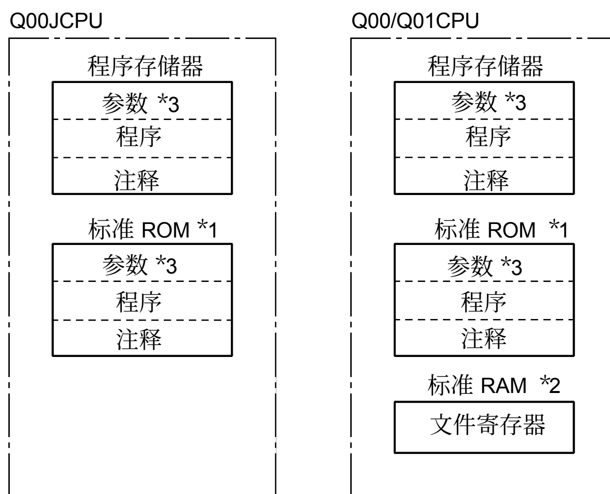
（10）利用文件口令防止非法存取

利用文件口令设置程序的存取级别（读保护、写保护），防止因不正当存取而引起程序的改变。

1.2　程序存储和运算

（1）程序的存储

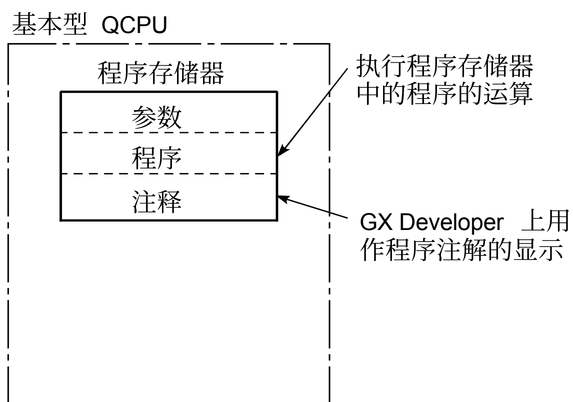
用GX Developer编写的程序可以存储在基本型QCPU的程序存储器、标准ROM中。



＊1 ：标准ROM可以在参数、程序和注释的ROM化时使用。
＊2 ：标准RAM可以用作文件寄存器。
＊3 ：包括GX Configurator所设置的智能功能模块的智能参数。

（2）程序的执行

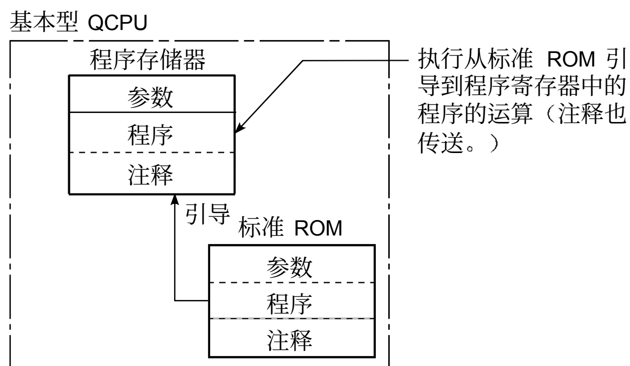
基本型QCPU对程序存储器中存储的程序进行运算。



（3）程序的引导运行

标准ROM中存储的程序引导（读出）到基本型QCPU的程序存储器后执行。

从标准ROM向程序存储器进行引导时，必须设置PC参数的引导文件 。

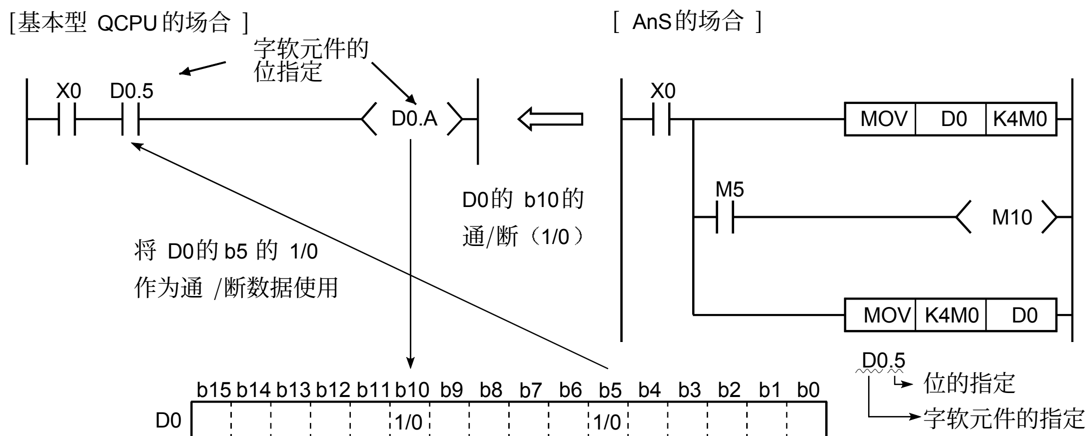


1.3 方便编程的软元件、指令

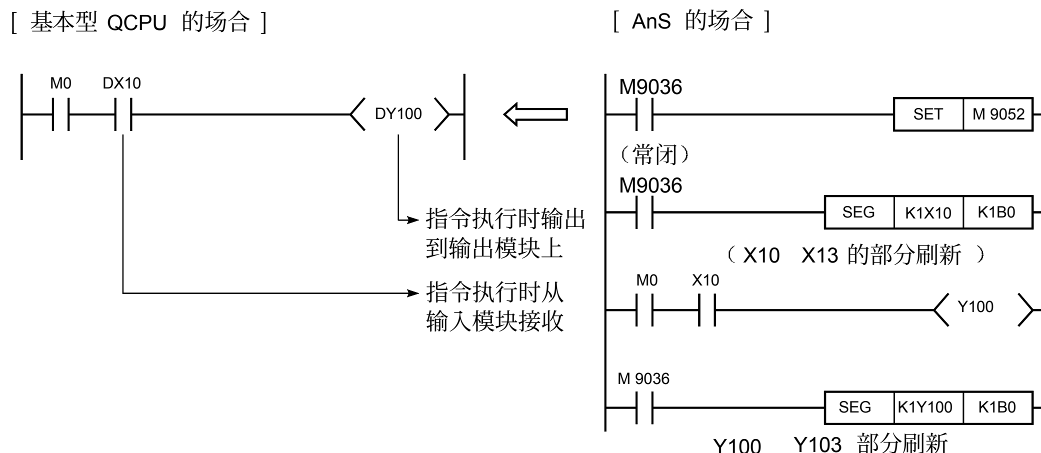
基本型QCPU上备有便于编写程序的软元件、指令，主要软元件、指令的概要如下所示。

（1）可以灵活地指定软元件

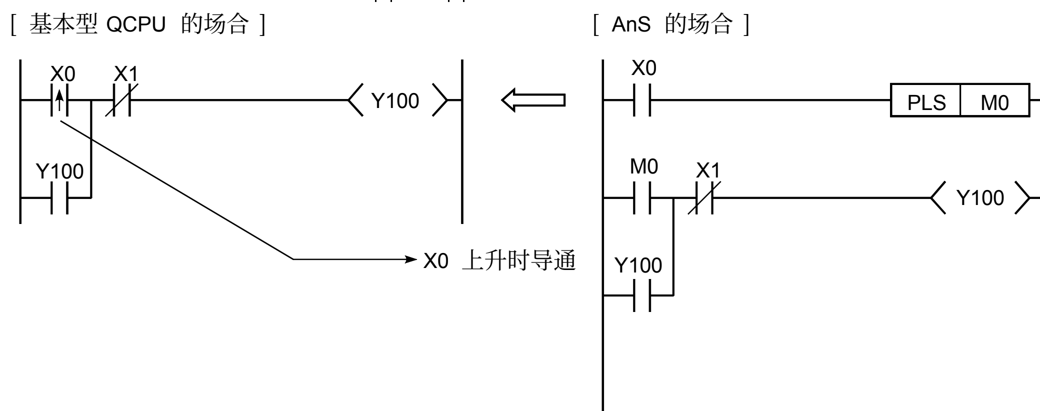
（a）通过字软元件的位指定，可以将字软元件的各个位作为触点、线圈处理。



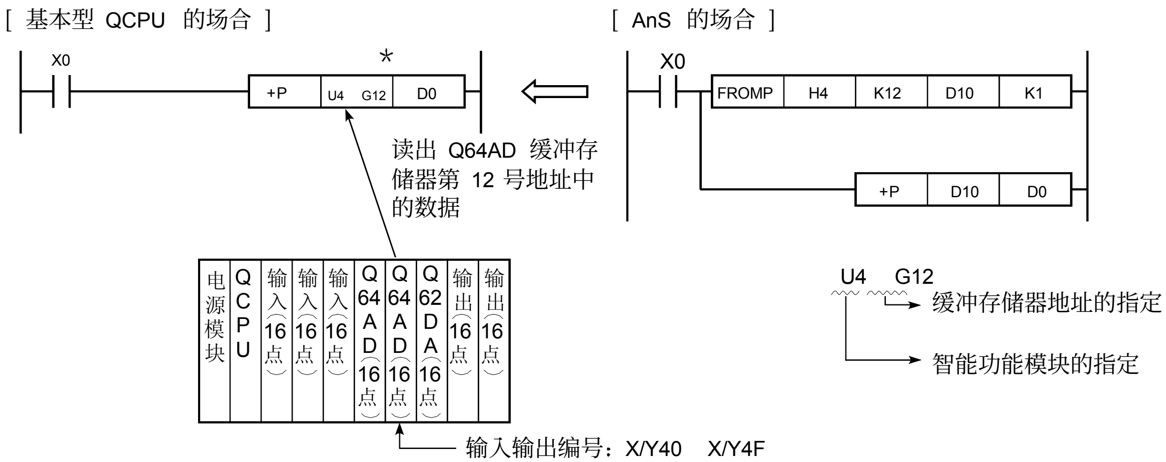
（b）利用直接存取输入（DX□）、直接存取输出（DY□），可以以1点单位方便地对程序直接进行处理。



（c）由于采用了微分触点（─┤↑├─ / ─┤↓├─），不再需要输入的脉冲化处理。

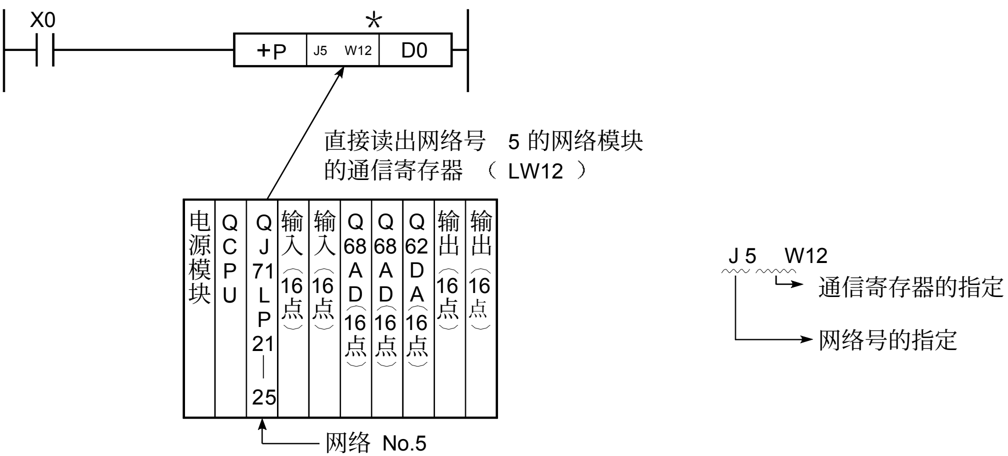


（d）编程时可以将智能功能模块（如：Q64AD、Q62DA）的缓冲存储器作软元件处理。



要　　点
通过用GX Configurator对所使用的智能功能模块进行自动刷新设置，可以不必直接对缓冲存储器进行存取，而以CPU的软元件寄存器为对象读出/写入相应的数据。

（e）MELSECNET/H网络模块（如：QJ71P21—25）的通信软元件（LX、LY、LB、LW、SB、SW）可以不必进行刷新设置就直接存取。

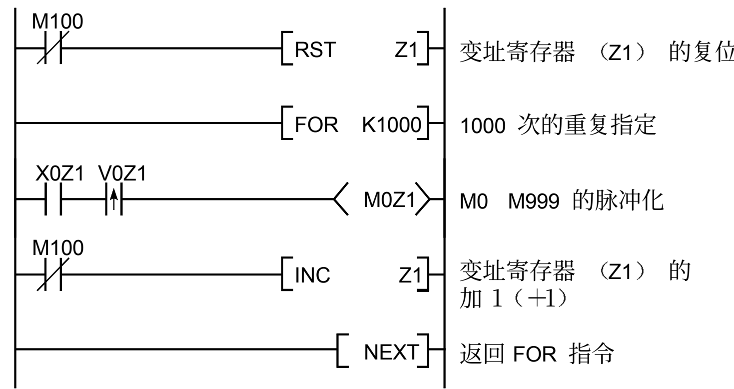


（f）基本型QCPU的各条指令即使进行变址修饰也不会延长处理时间，因此，可以方便地编写结构化的程序。

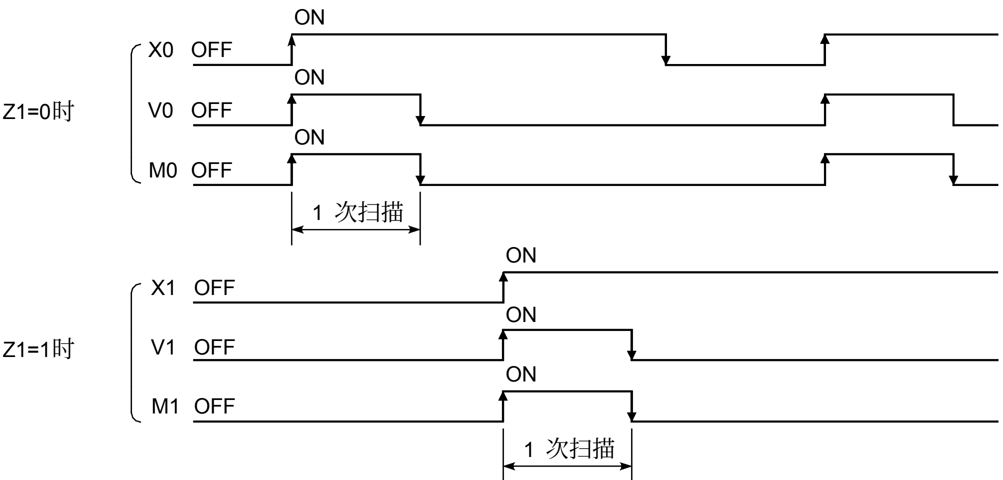
（2）通过边沿继电器的使用使脉冲处理变得容易

（a）通过采用输入条件上升时导通的边沿寄存器，使进行触点变址修饰情况下的脉冲处理变得容易。

[电路举例]

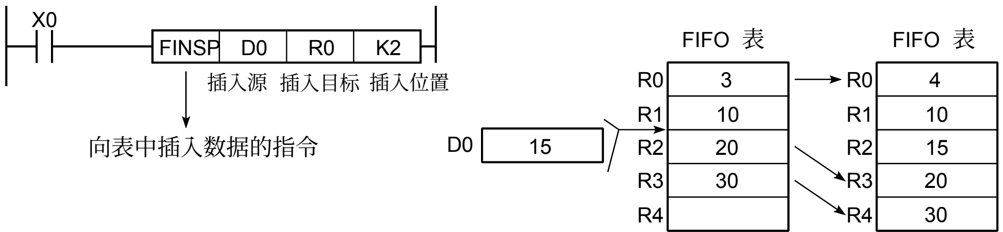


[时序图]

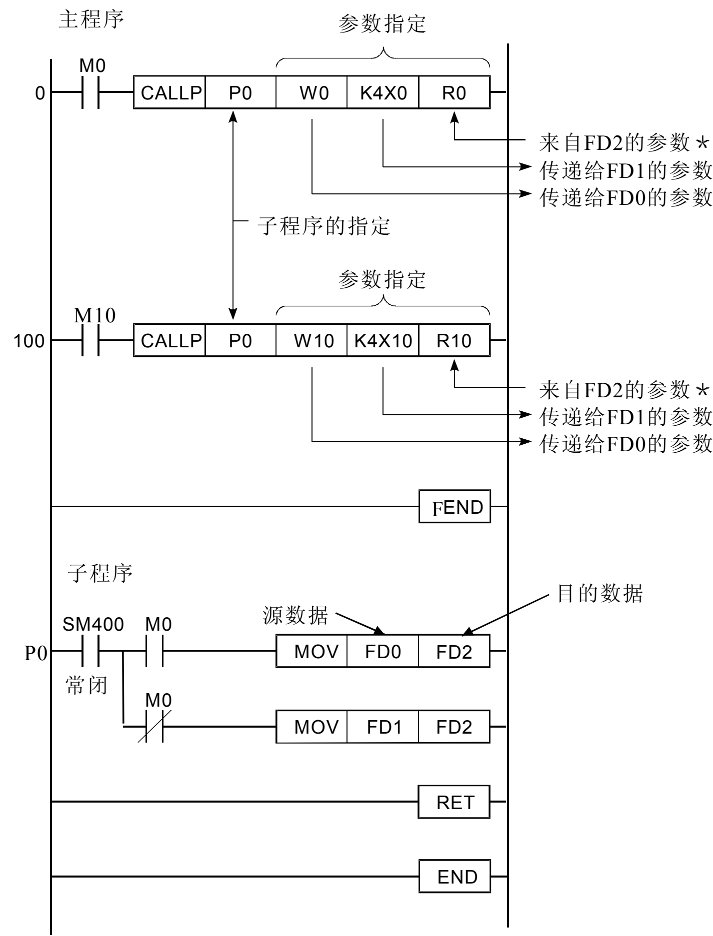


（3）简便的数据处理

通过充实表处理指令等数据处理，实现了大容量数据的高速处理。



（4）方便的子程序共享
利用带自变数的子程序调用指令，用于多次调用的子程序的编写变得容易。



备　注

*：关于参数的输入/输出条件，请参照10.3.1项。

第2章 系统构成

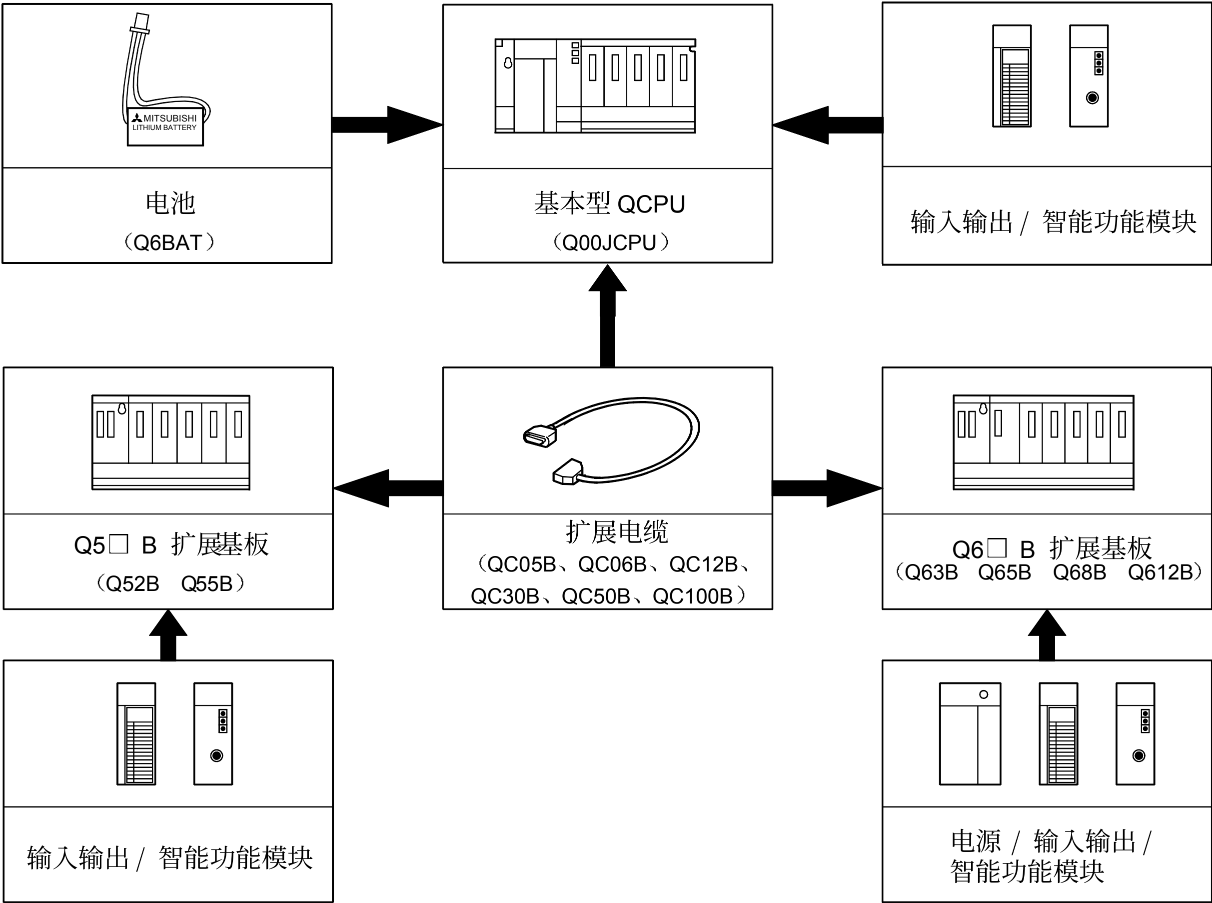
本章阐述基本型QCPU的系统构成、系统使用上的注意事项以及构成设备的有关内容。

2.1 系统构成

2.1.1 Q00JCPU的场合

本小节阐述Q00JCPU系统中的设备构成、系统构成的概要。

(1) 设备构成



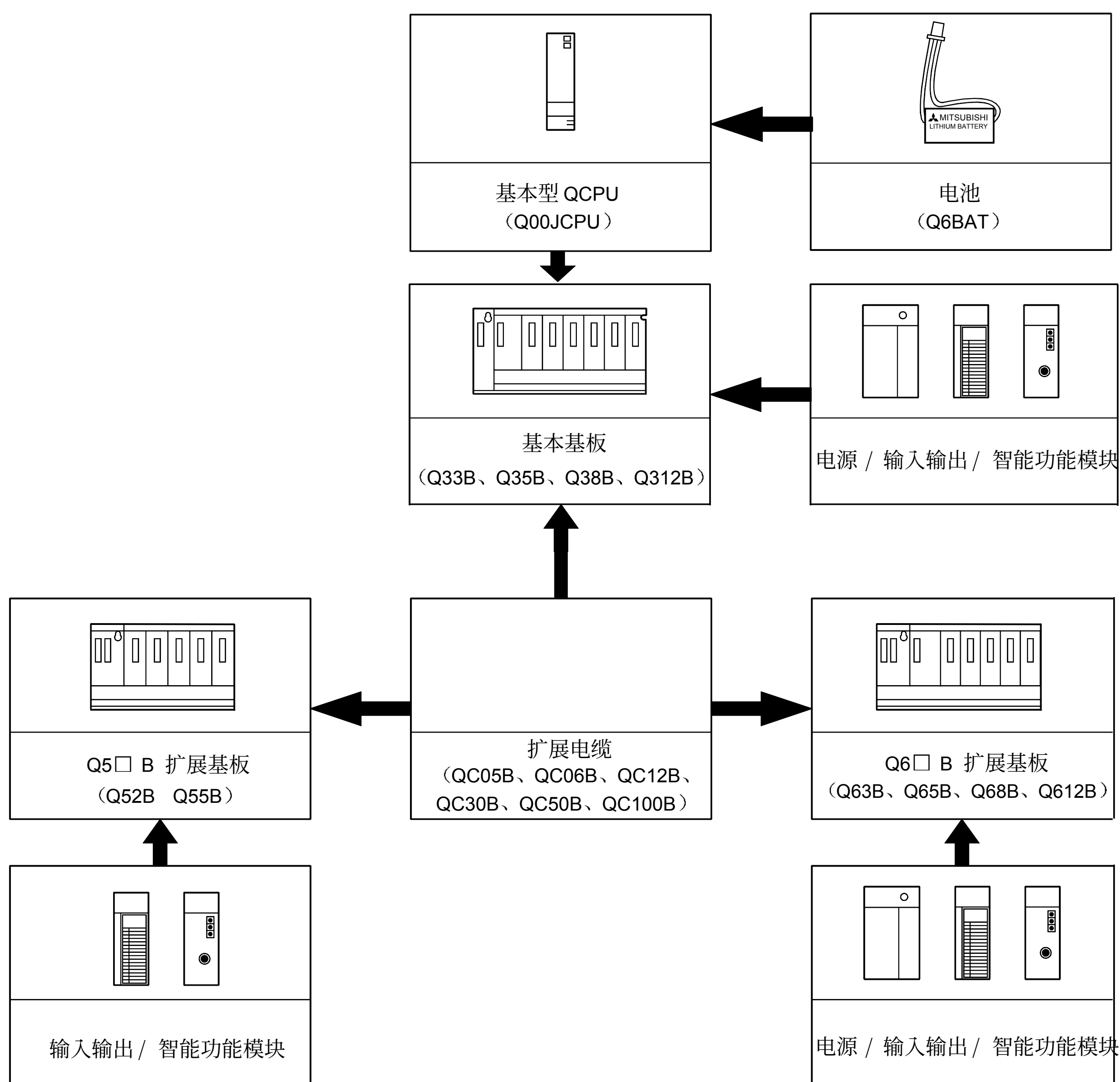
(2) 系统构成的概要

系统构成	(a) 连接了扩展基板的系统 (b) 连接了扩展基板和 GOT 的系统 扩展级数 ：2 插槽 No.:0 * 显示的是主基板和扩展基板的各插槽中安装了 16 点部件的场合。
最大扩展级数	扩展2级
最大输入输出模块安装数	10个模块
最大输入输出点数	256点
主基板型号名称	不要
扩展基板型号名称	Q52B、Q55B、Q63B、Q65B、Q68B、Q612B
扩展用电缆型号名称	QC05B、QC06B、QC12B、QC30B、QC50B、QC100B
注意事项	(1) 扩展电缆的总延长距离请控制在13.2 m以内。 (2) 扩展电缆请不要和主电路（高电压、大电流）的电线捆扎在一起，也不要靠近。 (3) 扩展级数的设置请采用升序，避免同一编号的重复使用。 (4) 作为扩展基板，不能连接QA1S6□B/QA65B。 (5) 扩展电缆请从基板的扩展电缆接插件的OUT连接到下一级的扩展基板的IN。 (6) 模块安装到17块以上时将会出错。 (7) GOT连接总线时，每扩展1级，就占用1个插槽。 (8) Q00JCPU将GOT作为16点的智能功能模块进行处理，因此，可以安装在基板上并进行控制的点数每台GOT将减少16点。 (9) Q00JCPU不可连接总线延长接插件（A9GT—QCNB）。请将总线延长接插件连接到扩展基板上。

2.1.2 Q00/Q01CPU的场合

本小节阐述Q00/Q01CPU系统中的设备构成、系统构成的概要。

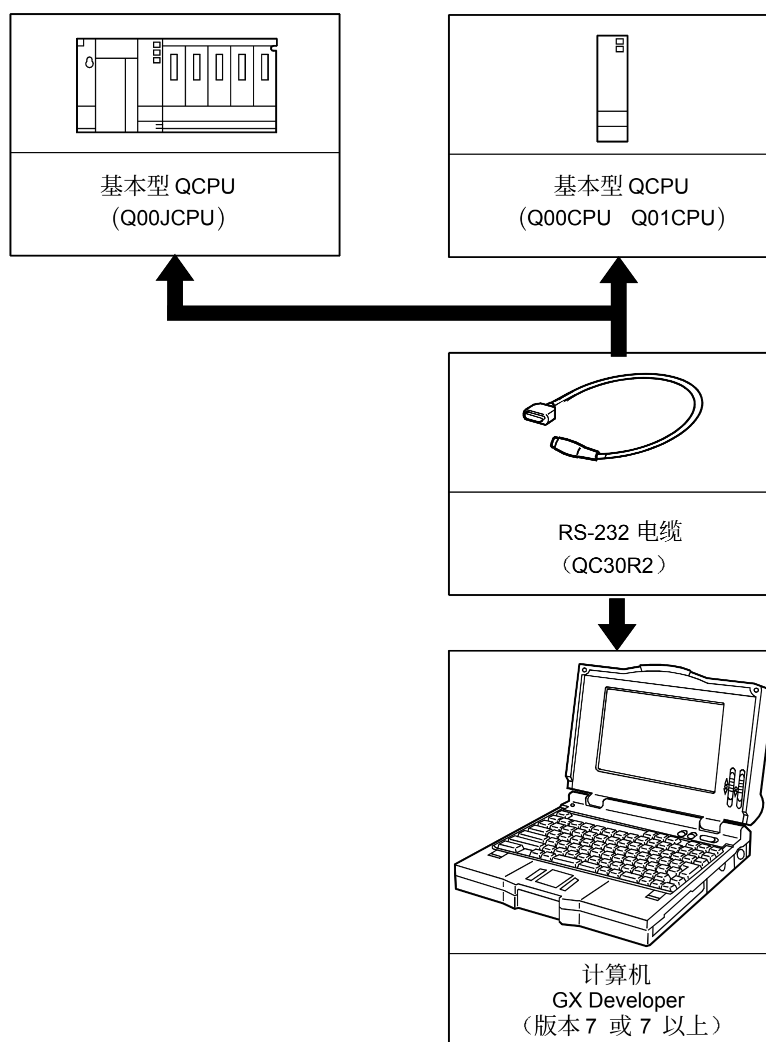
（1）设备构成



(2) 系统构成的概要

系统构成	基本基板（Q312B） 插槽号 0 1 2 3 4 5 6 7 8 9 10 11；OUT 电源模块 CPU 00–1F 20–3F 40–5F 60–7F 80–9F A0–BF C0–DF E0–FF 100–11F 120–13F 140–15F 160–17F 扩展电缆 增设 1 级　扩展基板（Q68B） 12 13 14 15 16 17 18 19；IN OUT 电源模块 180–19F 1A0–1BF 1C0–1DF 1E0–1FF 200–21F 220–23F 240–25F 260–27F 增设 2 级　扩展基板（Q65B） 20 21 22 23；IN OUT 电源模块 280–29F 2A0–2BF 2C0–2DF 2E0–2FF 禁止 → 安装时出错 ＊ 显示的是各插槽中安装了 32 点模块的场合。
最大扩展级数	扩展4级
最大输入输出模块安装数	24个模块
最大输入输出点数	1024点
主基板型号名称	Q33B、Q35B、Q38B、Q312B
扩展基板型号名称	Q52B、Q55B、Q63B、Q65B、Q68B、Q612B4
扩展用电缆型号名称	QC05B、QC06B、QC12B、QC30B、QC50B、QC100B
注意事项	(1) 扩展电缆的总延长距离请控制在13.2 m以内。 (2) 扩展电缆请不要和主电路（高电压、大电流）的电线捆扎在一起，也不要靠近。 (3) 扩展级数的设置请采用升序，避免同一编号的重复使用。 (4) 作为扩展基板，不能连接QA1S6□B/QA65B。 (5) 扩展电缆请从基板的扩展电缆接插件的OUT连接到下一级的扩展基板的IN。 (6) 模块安装到25块以上时将会出错。 (7) GOT连接总线时，每扩展1级，就占用1个插槽。 (8) Q00/Q01CPU将GOT作为16点的智能功能模块进行处理，因此，可以安装在基板上并进行控制的点数每台GOT将减少16点。

2.1.3 GX Developer的构成



2.2 使用上的注意事项

本节阐述基本型QCPU上可以使用的硬件和软件。

(1) 硬件

(a) 根据模块的种类，安装台数和功能有所限制。

品名	型号名称	安装块数/限制
Q系列MELSECNET/H 网络模块	• QJ71LP21 • QJ71BR11 • QJ71LP21-25 • QJ71LP21G	仅1块
Q系列Ethernet（以太网） 接口模块	• QJ71E71 • QJ71E71-B2 • QJ71E71-100	仅1块
Q系列CC—Link系统 主站 • 本地站模块	• QJ61BT11	最多2块 功能版本B以后的产品可以使用。
中断模块	• QI60	仅1块

(b) 图形操作终端只有在GOT900系列、F900系列（需要安装Q模式对应基本操作系统（OS）和通信驱动程序）上可以使用。
GOT800系列、A77GOT、A64GOT不可使用。

(c) FL—net模块（QJ71FL71、QJ71FL71—B2)在功能版本B以后的产品上可以用于基本型QCPU 。

(2) 软件包

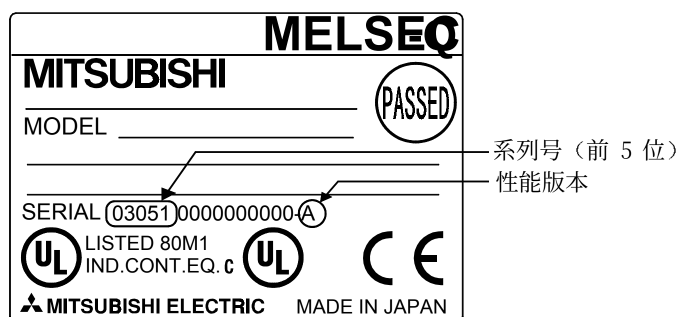
GX Developer、GX Configurator 在下表的功能版本及以后的产品上可以用于基本型QCPU。

产品名称	型号名称	版本
GX DEVELOPER	SW7D5C-GPPW	Ver.7
GX 模拟器	SW6D5C-LLT	Ver.6
GX Configurator-AD	SW0D5C-QADU	Ver.1.10L
GX Configurator-DA	SW0D5C-QDAU	Ver.1.10L
GX Configurator-SC	SW0D5C-QSCU	Ver.1.10L
GX Configurator-CT	SW0D5C-QCTU	Ver.1.10L
GX Configurator-TC	SW0D5C-QTCU	Ver.1.10L
GX Configurator-FL	SW0D5C-QFLU	Ver.1.10L
GX Configurator-QP	SW2D5C-QD75P	Ver.2.10L

2.3 功能版本的确认方法

基本型QCPU的功能版本可以采用额定铭牌和GX Developer的系统监视进行确认。

（1）查看额定铭牌
铭牌上可以进行功能版本的确认。



（2）系统监视（产品信息一览）下的功能版本的确认
在GX Developer的系统监视的产品信息一览中 ，可以对基本型QCPU的功能版本进行确认。
而且，系统监视的产品信息一览中，也可以对智能功能模块的功能版本进行确认。



ｽﾛｯﾄ	種別	ｼﾘｰｽﾞ	形名	点数	先頭I/O	管理CPU	ｼﾘｱﾙNo	Ver
CPU	CPU	Q	Q01CPU	-	-	-	030510000000000	A
0-0	-	-	空き	-	-	-	-	-
0-1	-	-	空き	-	-	-	-	-
0-2	-	-	空き	-	-	-	-	-
0-3	-	-	空き	-	-	-	-	-
0-4	-	-	空き	-	-	-	-	-
0-5	-	-	空き	-	-	-	-	-
0-6	-	-	空き	-	-	-	-	-
0-7	-	-	空き	-	-	-	-	-

第3章 性能要求

基本型QCPU的性能要求的有关内容如下所示。

表3.1 性能规格

项目		型号名称			备注
		Q00JCPU	Q00CPU	Q01CPU	
控制方式		存储程序的反复运算			
输入输出控制方式		刷新方式			通过直接存取输入输出（DX、DY）的指定，可以只进行直接的输入输出
编程语言（可编程控制器控制专用语言）		继电器符号语言 逻辑符号语言			不对应SFC功能。
处理速度（顺控指令）	LD X0	0.20μs	0.16μs	0.10μs	
	MOV D0 D1	0.70μs	0.56μs	0.35μs	
指令总数		249条（智能功能模块专用指令除外）			
恒定扫描（保持扫描时间恒定的功能）		2～2000 ms（可以以1 ms为单位进行设置）			由参数设置
程序容量∗1 ∗2		8k步（32k字节）	8k步（32k字节）	14k步（56k字节）	
存储容量	程序存储器（驱动0）	58k字节	94k字节	94k字节	参照6.2节
	标准RAM（驱动3）	0	64k字节	64k字节	参照6.5节
	标准ROM（驱动4）	58k字节	94k字节	94k字节	参照6.3节
文件最大存储数	程序存储器	1	1	1	参照6.2节
	标准ROM	1	1	1	参照6.3节
存储文件寄存器数	标准RAM	——	1	1	参照6.5节
输入输出软元件点数		2048点(X/YO～7FF)			程序上可以使用的点数
输入输出点数		256点(X/YO～FF)	1024点(X/YO～3FF)		可以与输入输出模块存取的点数

∗1：程序容量的1步为4个字节。
∗2：可执行的最大顺控步数如下式所示，即：
（程序容量）—（文件头的大小（默认值：34步））

表3.1 性能规格（续）

<table>
<tr><th colspan="3" rowspan="2">项目</th><th colspan="3">型号名称</th><th rowspan="2">备注</th></tr>
<tr><th>Q00JCPU</th><th>Q00CPU</th><th>Q01CPU</th></tr>
<tr><td rowspan="23">软元件点数</td><td colspan="2">内部继电器[M]</td><td colspan="3">默认值8129点（M0～8191）</td><td rowspan="10">通过参数在16.4k字的范围内设置使用点数</td></tr>
<tr><td colspan="2">锁存继电器[L]</td><td colspan="3">默认值2048点（L0～2047）</td></tr>
<tr><td colspan="2">通信继电器[B]</td><td colspan="3">默认值2048点（B0～7FF）</td></tr>
<tr><td colspan="2">定时器[T]</td><td colspan="3">默认值512点（T0～511）（低速定时器/高速定时器共用）
低速定时器/高速定时器的转换采用指令设置
低速定时器/高速定时器的计测单位由参数设置
（低速定时器：1～1000 ms，1 ms单位，默认值100 ms）
（高速定时器：0.1～100 ms，0.1 ms单位，默认值10 ms）</td></tr>
<tr><td colspan="2">累加定时器[ST]</td><td colspan="3">默认值0点（低速保持定时器/高速保持定时器共用）
低速累加定时器/高速保持定时器的转换采用指令设置
低速累加定时器/高速保持定时器的计测单位由参数设置
（低速累加定时器：1～1000 ms，1 ms单位，默认值100 ms）
（高速累加定时器：0.1～100 ms，0.1 ms单位，默认值10 ms）</td></tr>
<tr><td colspan="2">计数器[C]</td><td colspan="3">普通计数器　默认值512点（C0～511）
中断计数器　最大128点（默认值0点，由参数设置）</td></tr>
<tr><td colspan="2">数据寄存器[D]</td><td colspan="3">默认值11136点（D0～11135）</td></tr>
<tr><td colspan="2">通信寄存器[W]</td><td colspan="3">默认值2048点（W0～7FF）</td></tr>
<tr><td colspan="2">信号报警器[F]</td><td colspan="3">默认值1024点（F0～1023）</td></tr>
<tr><td colspan="2">边沿继电器[V]</td><td colspan="3">默认值1024点（V0～1023）</td></tr>
<tr><td rowspan="2">文件寄存器</td><td>[R]</td><td>无</td><td colspan="2">32768点（R0～32767）</td><td rowspan="13">软元件点数固定</td></tr>
<tr><td>[ZR]</td><td>无</td><td colspan="2">32768点（ZR0～32767）</td></tr>
<tr><td colspan="2">特殊通信继电器[SB]</td><td colspan="3">1024点(SB0～3FF)</td></tr>
<tr><td colspan="2">特殊通信寄存器[SW]</td><td colspan="3">1024点(SW0～3FF)</td></tr>
<tr><td colspan="2">步进继电器[S] ＊3</td><td colspan="3">2048点(S0～2047)</td></tr>
<tr><td colspan="2">变址寄存器[Z]</td><td colspan="3">10点(Z0～9)</td></tr>
<tr><td colspan="2">指针[P]</td><td colspan="3">300点(P0～299)</td></tr>
<tr><td colspan="2">中断指针[I]</td><td colspan="3">128点(I0～127)
可以通过参数设置系统中断的I28～I31的周期间隔（2～100 ms，1 ms单位）
默认值I28：100 ms　I29：40 ms　I30：20 ms I31：10 ms</td></tr>
<tr><td colspan="2">特殊继电器[SM]</td><td colspan="3">1024点(SM0～1023)</td></tr>
<tr><td colspan="2">特殊寄存器[SD]</td><td colspan="3">1024点(SD0～1023)</td></tr>
<tr><td colspan="2">功能输入[FX]</td><td colspan="3">16点(FX0～F)</td></tr>
<tr><td colspan="2">功能输出[FY]</td><td colspan="3">16点(FY0～F)</td></tr>
<tr><td colspan="2">功能寄存器[FD]</td><td colspan="3">5点(FD0～4)</td></tr>
</table>

＊3：步进继电器是SFC功能用的软元件。
由于基本型QCPU不对应SFC功能，因此不可使用。

表3.1 性能规格（续）

项目		型号名称			备注
		Q00JCPU	Q00CPU	Q01CPU	
直接通信软元件		直接存取通信软元件的软元件。 MELSECNET/H专用。 指定形式：J□□¥X□□、J□□¥Y□□、J□□¥W□□、 J□□¥B□□、J□□¥SW□□、J□□¥SB□□			
智能功能模块软元件		直接存取智能功能模块的缓冲存储器的软元件。指定形式：U□□¥G□□			
锁存（停电保持）		L0～2047（默认值） （可以对B、F、V、T、ST、C、D、W设置锁存范围）			通过参数设置
远程运行/暂停触点		可以从X0～7FF中各设置1个运行/暂停触点			
时钟功能		年、月、日、时、分、秒、星期（自动判别闰年） 精度 —3.2～＋5.27 s（TYP.＋1.98 s）/d 0℃时 精度 —2.57～＋5.27 s（TYP.＋2.22 s）/d 25℃时 精度 —11.68～＋3.65 s（TYP.—2.64 s）/d 55℃时			
允许瞬间掉电时间		20 ms以内 （AC100V以上）	根据电源模块决定		
DC5V内部消费電流		0.22A ＊4	0.25A	0.27A	
质量		0.66kg ＊5	0.13kg		
外形尺寸	H	98mm	98mm		
	W	245mm ＊5	27.4mm		
	D	97mm	89.3mm		

＊4：包括CPU模块、基板的值。
＊5：包括CPU模块、基板、电源模块的值。

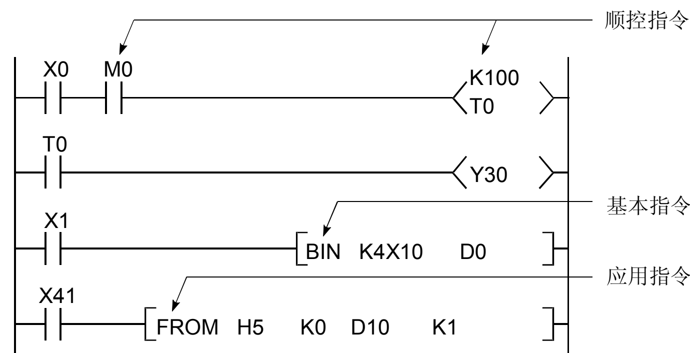
第4章 顺控程序的构成和执行条件

基本型QCPU上可以执行的程序只有顺控程序。
本章将阐述顺控程序的构成和执行条件的有关内容。

4.1 顺控程序

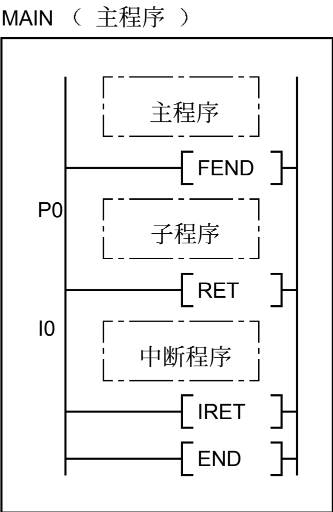
（1）什么是顺控程序

（a）顺控程序就是采用顺控指令、基本指令、应用指令等编写而成的程序。



（b）顺控程序可分类为主程序、子程序和中断程序3种程序。
主程序、子程序、中断程序的详细内容请参照以下章节。

- 主程序：4.1.1项
- 子程序：4.1.2项
- 中断程序：4.1.3项



备 注

顺控指令、基本指令、应用指令的有关内容请参照下列手册。

- QCPU（Q模块）/QnACPU编程手册（通用指令篇）

4

（2）顺控程序的记述方法

顺控程序的编程方法有2种，即采用电路模式的方法和采用列表模式的方法。

（a）电路模式

- 电路模式以继电器控制的顺控电路为基本思路，可以采用接近顺控电路的表现形式进行编程。
- 采用电路模式的编程以电路块为单位进行。

电路块是进行顺控程序运算的最小单位，是从左侧的纵向母线开始的，到右侧的纵向母线为止的电路。

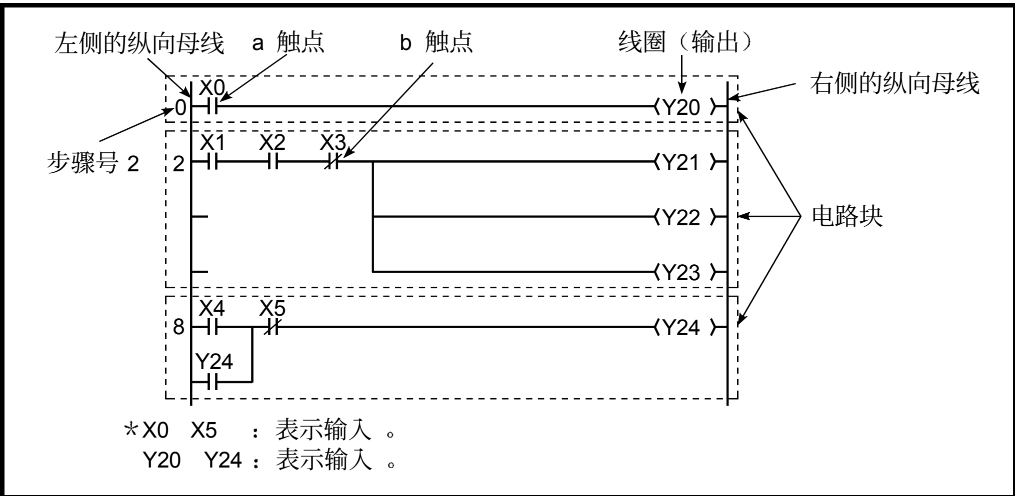


图4.1 电路块

（b）列表模式

列表模式采用专用指令，对电路模式中使用记号形式的触电、线圈等进行编程。

a触点、b触点、线圈的指令分别如下：

- a触点 • • • • • LD、AND、OR
- b触点 • • • • • LDI、ANI、ORI
- 线圈 • • • • • OUT

（2）程序的运算

顺控程序的运算从第0步开始，直到END/FEND为止依次执行。

电路模式的电路块从左侧的纵向母线向右侧，从上到下进行运算。

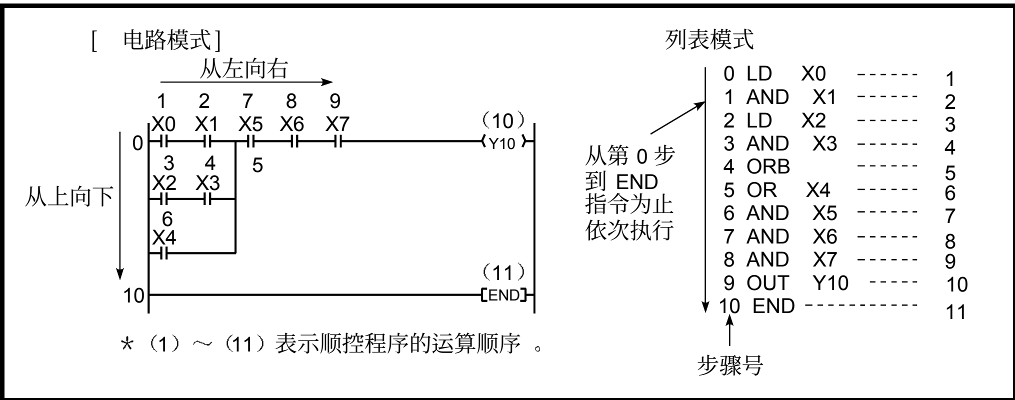


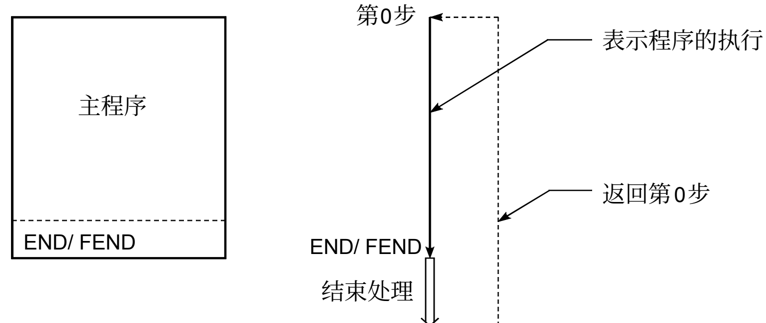
图4.2 顺控程序的运算

4.1.1 主程序

（1）什么是主程序

（a）主程序就是从第0步开始到END/FEND指令为止的程序。＊1

（b）主程序从第0步开始一直执行到END/FEND指令为止。
一旦执行主程序的END/FEND指令，就在进行结束处理后，再次从第0步开始进行运算。



（2）主程序的执行
主程序每次扫描时执行。

备　注

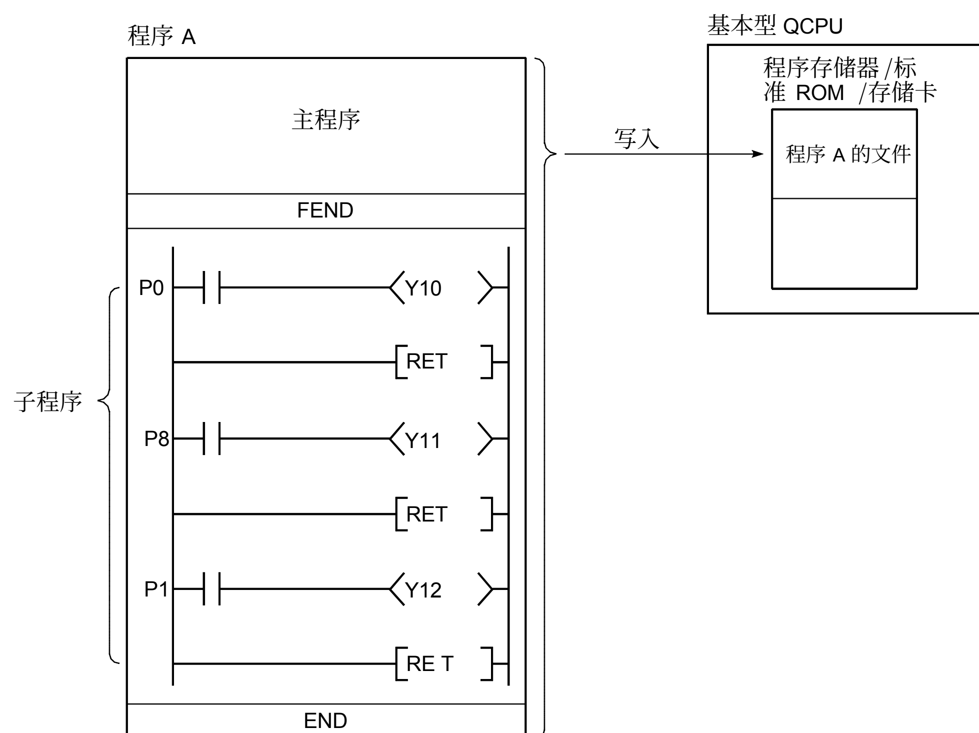
*1: END/FEND指令的详细内容请参照下列手册。
- QCPU（Q模块）/QnACPU编程手册（通用指令篇）

4.1.2 子程序

(1) 什么是子程序
(a) 子程序就是从指针（P□）开始到RET指令为止的程序。
(b) 子程序仅当由来自主程序的子程序调用指令（CALL（P）、FCALL（P）等）调用时才执行。
(c) 子程序的用途
① 将1次扫描中多次执行的程序编成子程序，可以减少整体步数。
② 将仅当某个条件成立时才要求执行的程序编成子程序，可以减少平时执行的程序步数。

(2) 子程序的管理
子程序编在主程序之后（FEND以后），作为1个程序进行管理。

子程序的编写方法如下。
- 在主程序的FEND指令～END指令之间编写子程序。
- 子程序的编写次序没有限制。
  而且，编写多个子程序时，没有必要将指针按新编号顺序排列。



4.1.3 中断程序

（1）什么是中断程序

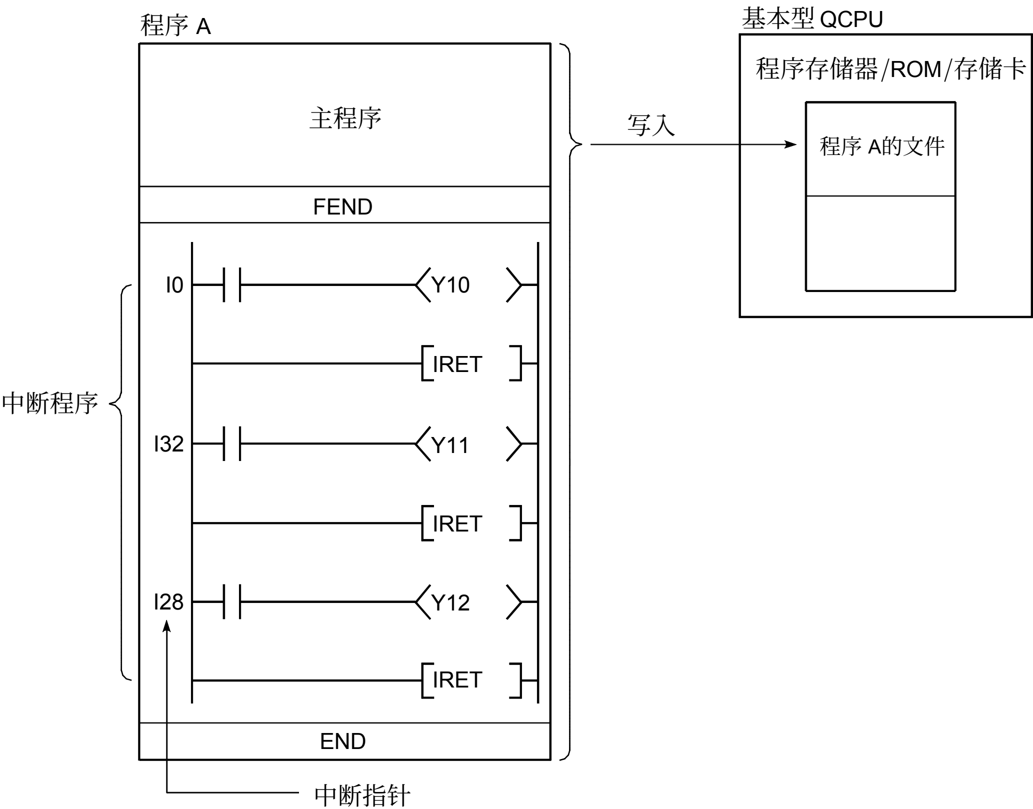
（a）中断程序就是从中断指针（I□）到IRET指令为止的程序。 ＊1

（b）中断程序仅当中断原因发生时才执行。＊1

（2）中断程序的管理

中断程序编在主程序之后（FENG指令以后），作为1个程序进行管理。
中断程序的编写方法如下。

- 在主程序的FEND指令～END指令之间编写中断程序。
- 中断程序的编写次序没有限制。
  而且，编写多个中断程序时，没有必要将指针按新编号顺序排列。



备　注

＊1：中断的原因、中断指针的详细内容请参照10.10节。

（3）中断程序的执行

（a）中断程序执行时，必须利用EI指令先设置为中断允许状态。＊1

① 在进入中断允许状态之前发生中断原因的情况下，记忆所发生的中断原因，待进入中断允许状态之时，再执行与中断原因相应的中断程序。

② 多次发生同一中断原因的情况下，所发生的中断原因被记忆或丢弃。

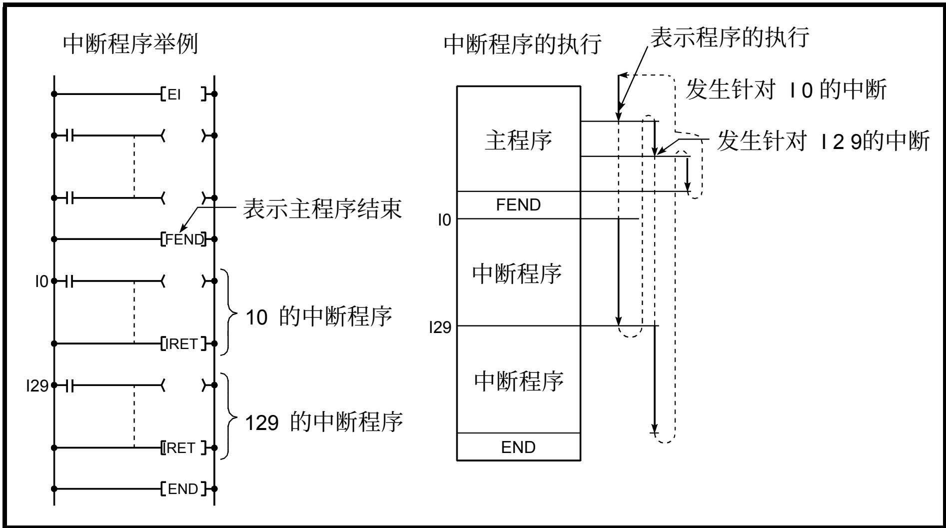


图4.3 中断程序的执行

（b）一旦发生中断原因，与该原因相应的中断指针编号的中断程序就被执行。

但是，根据所要执行的中断时的条件，中断程序的执行会有所不同。

① 同时发生多个中断的场合

多个中断程序同时发生中断的情况下，优先级别较高的中断指针编号（I□）所对应的中断程序将被执行。＊2

其他中断程序在执行中的中断程序的处理完毕之前将被迫等待。

执行中的中断程序的处理完成之前，又发生与执行中的中断程序同一中断原因的中断时，将所发生的中断原因记忆或丢弃。

② 指令执行中的场合

各条指令的执行过程中为中断禁止状态。

各条指令的执行过程中发生中断时，要在正在执行的指令处理完毕之时再执行中断程序。

③ 网络刷新中的中断
如果网络刷新期间发生中断，就中断网络刷新，执行中断程序。
因此，在MELSECNET/H网络系统上，即便进行“循环数据的站单位块保证”，一旦在中断程序中使用刷新前已设置的软元件，就不能实现上述保证，因此，请加以注意。*3

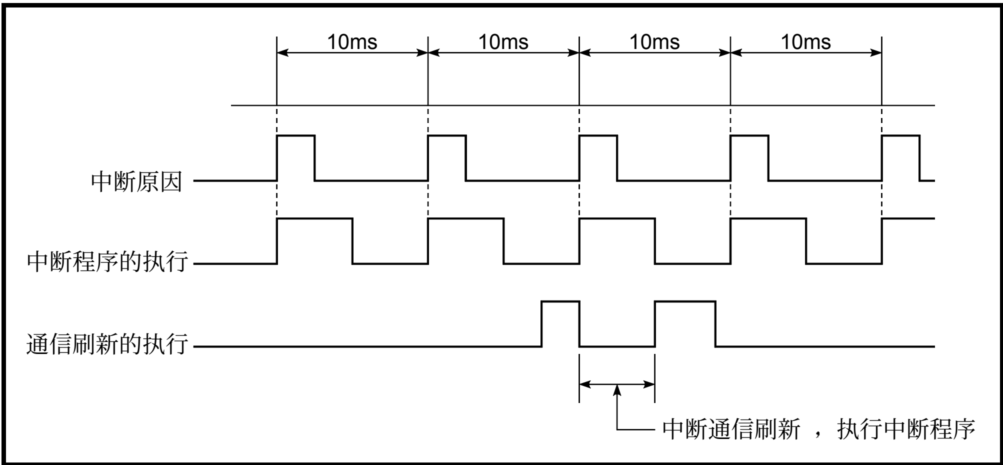


图4.4 网络刷新中的中断状态

④ 结束处理中的中断
执行恒定扫描的过程中，在结束处理的等待时间中发生中断原因的情况下，执行与中断原因相对应的中断程序。

（c）关于从主程序转换到中断程序时的变址寄存器的处理，请参照10.6.1项。

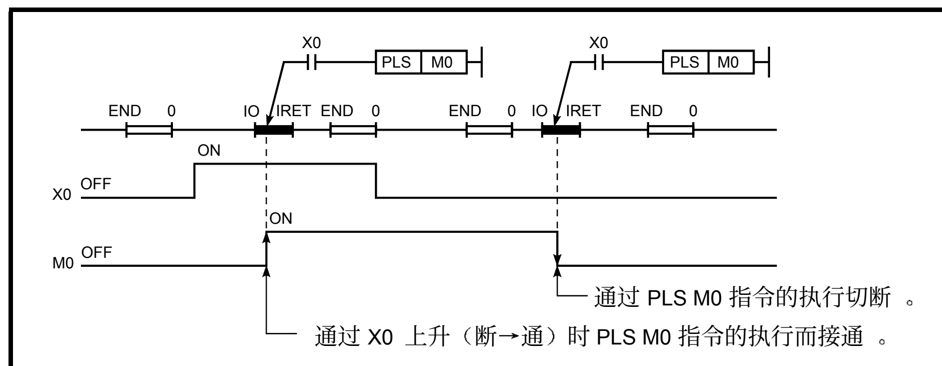
（4）中断程序高速执行的设置和辅助处理时间
基本型QCPU的默认值设置为执行中断程序的情况下，进行“变址寄存器的保存和恢复”。
在PC参数的PC系统设置时，如果选择中断程序的“高速执行”，就不进行上述处理。
这样，就可以缩短中断程序的辅助处理时间。
中断程序的辅助处理时间请参照11.2节。

备 注

*1：关于EI指令，请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）
*2：关于中断程序的优先级别，请参照10.10节。
*3：关于循环数据的站单位块保证，请参照下列手册。
• Q对应MELSECNET/H网络系统参考手册

(5) 编程时的限制

(a) 由中断程序内的PLS指令接通的软元件在同一软元件的PLS指令再次执行前将保持"通"的状态。



(b) 中断程序的执行过程中，为了不执行其他中断处理，处于中断禁止（DI）状态。
中断程序中请不要执行EI/DI指令。

(c) 中断程序中不可使用定时器。
定时器在OUT T□指令执行时进行当前值的更新和触点的通/断，因此，如果在中断程序中使用定时器，将会出现只在中断程序执行完毕时更新当前值的情况，以致无法再进行正常的计测。

(d) 中断程序中不可使用以下指令。
- COM
- ZCOM
- EI
- DI

(e) 扫描时间等的时间计测时，如果执行中断程序，所计测的时间值将加上中断程序的执行时间。
这样，一旦执行中断程序，以下的特殊寄存器的保存值和GX Developer的监视值将比没有执行中断程序时来得要长。

① 特殊寄存器
- SD520、SD521：当前的扫描时间
- SD524、SD525：最小扫描时间
- SD526、SD527：最大扫描时间
- SD540、SD541：结束处理时间
- SD542、SD543：恒定扫描等待时间
- SD548、SD549：扫描程序执行时间

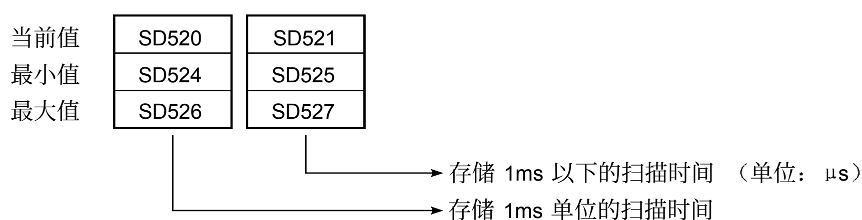
② GX Developer的监视值
- 扫描时间的测定
- 恒定扫描

4.2 扫描时间的设想

（1）扫描时间

（a）扫描时间为顺控程序的执行时间和结束处理时间的合计时间。
执行中断程序的情况下，扫描时间为加上中断程序执行时间的值

（b）在基本型QCPU上，对扫描时间的当前值、最小值、最大值进行计测，并存储在特殊寄存器（SD520、SD521、SD524～SD527）中。通过对SD520、SD521、SD524～SD527的监视，可以确认扫描时间。



SD520为3，SD521为4的情况下，扫描时间为3.4 ms。

要　点
＊1：特殊寄存器内存储的各扫描时间的精度为±0.1 ms。 此外，即使在顺控程序中执行监视时间复位指令（WDT），各扫描时间的计测也会继续。

（2）设置为恒定扫描时＊2

设置为恒定扫描时，设置的每个恒定扫描时间执行主程序。

（3）WDT（监视时间）

用于监视扫描时间的定时器的默认值设置为200 ms。
WDT在PC参数的PC RAS设置时可以在10 ms～2000 ms的范围内变化。
（设置单位：10 ms）

要　点
WDT的计测误差为10 ms。 因此，如果将WDT（t）设置为10 ms，扫描时间就处于 10 ms＜t＜20 ms的范围内，出现WDT出错。

备　注

＊2：恒定扫描是以一定间隔反复执行主程序的功能。
有关恒定扫描的详细内容请参照7.2节。

4.3 运算处理

4.3.1 初始化处理

初始化处理是为执行顺控运算而进行的预处理，仅在下表的情况下执行1次。
初始化处理完毕后，基本型QCPU就处于用运行/停止/复位开关所设置的状态。（参照4.4节）

初始化处理项目	基本型QCPU的状态		
	电源打开时	复位操作时	停止状态→运行状态时＊1
输入输出模块的初始化	○	○	×
来自标准ROM的引导	○	○	×
锁存范围外的软元件的初始化（位软元件：断，字软元件：0）	○	○	×
自我诊断的执行	○	○	×
安装模块的输入输出编号的自动分配	○	○	×
MELSECNET/H网络信息的设置和网络通信的开始	○	○	×
智能功能模块的开关设置的设置	○	○	×
CC—Link信息的设置	○	○	×
Ethernet信息的设置	○	○	×
实时通信功能的设置	○	○	×

○：执行、 ×：不执行

备　注

＊1：在停止状态下更改参数或程序时，请用运行/停止/复位开关进行复位操作。
如果不进行复位，而对运行/停止/复位开关进行停止→运行转换，则运行LED（发光二极管）将会闪烁。
再次对运行/停止/复位开关进行运行→停止→运行转换，就进入运行状态，“停止状态→运行状态时”的动作将会有效。
但是，脉冲化指令（PLS，□P）因上一次的信息根据程序的变更内容将不再继续，有可能不能正常动作，请充分注意。

4.3.2 输入/输出刷新（输入输出模块的刷新处理）

输入/输出刷新时，将从输入模块/智能功能模块接收输入（X），并将基本型QCPU运算出的输出（Y）输出到输出模块/智能功能模块上。

输入/输出刷新在顺控程序的运算开始前执行。
此外，执行恒定扫描时，要等恒定扫描的等待时间结束后才会进行输入/输出刷新。（输入/输出刷新每个恒定扫描时间会执行1次。）

4.3.3 智能功能模块的自动刷新

已设置智能功能模块的自动刷新的情况下，与智能功能模块就所设置的数据进行通信。智能功能模块的自动刷新设置的有关内容请参照所使用的智能模块的手册。

4.3.4 结束处理

结束处理就是结束1次扫描的顺控程序的运算处理，使顺控程序的执行返回到第0步的后处理。

结束处理有以下几种处理：
- MELSECNET/H、CC—LINK的刷新处理
- 智能功能模块的自动刷新
- 自我诊断
- 与GX Developer等的外部设备之间的通信
- 智能功能模块专用指令的处理

4.4 运行状态、停止状态、暂停状态的运算处理

基本型QCPU的动作状态有3种，即运行状态、停止状态、暂停状态。
本节阐述各种动作状态时的基本型QCPU的运算处理的有关内容。

（1）运行状态的运算处理

（a）所谓运行状态，就是顺控程序的运算依照第0步→END（FEND）→第0步的顺序，反复执行运算的状态。

（b）进入运行状态时，根据停止→运行时的输出模式的参数设置，将停止状态时保存的输出状态输出，或将1次扫描后的运算结果输出。

（c）从停止→运行转换后到顺控程序运算开始之前的处理时间根据系统构成有所变化，通常为1～3秒。
但是，根据条件的不同，有时也可能延长。

（2）停止状态的运算处理

（a）所谓停止状态，就是利用运行/停止/RESRT开关或远程停止功能中止顺控程序运算的状态。（远程停止功能的有关内容请参照7.6.1项。）
此外，发生停止出错的情况下，也进入停止状态。

（b）进入停止状态时，保存输出状态，切断所有输出点。
输出（Y）以外的软软元件寄存器将被保持。

（3）暂停状态的运算处理

所谓暂停状态，就是利用远程暂停功能，在保持输出及软元件寄存器状态的情况下，中止顺控程序运算的状态。
（远程暂停功能的有关内容请参照7.6.2项。）

（4）运行/停止/复位开关操作时基本型QCPU的运算处理

运行/停止状态 ＼ 运算处理	顺控程序的运算处理	外部输出	软软元件寄存器（M、L、S、T、C、D）
运行→停止	执行到END指令为止后停止。	保存输出状态，切断所有输出点。	保持即将进入停止状态之前的状态。
停止 →运行	从第0步开始	根据PC参数从停止 →运行时的输出模式决定	从即将进入停止状态之前的状态执行运算。

要　点
基本型QCPU不管处于运行状态、停止状态、暂停状态中的哪个状态，都在进行以下处理。 ・与输入输出模块之间的刷新处理 ・与GX Developer、串行通信模块之间的数据通信 ・MELSECNET/H、CC—Link的刷新处理 因此，即使设置为停止状态或暂停状态，也可以采用GX Developer进行输入输出监视和测试操作，从串行通信读出/写入，利用MELSECNET/H与其他站通信，或与CC—Link的远程站通信。

4.5 瞬间掉电时的运算处理

基本型QCPU在供给电源模块的输入电源电压低于规定范围时，可检测出瞬间掉电。
基本型QCPU检测出瞬间掉电时，进行以下的运算处理。

（1）发生允许瞬间掉电时间以内的瞬间掉电时
　（a）发生瞬间掉电时，保持输出状态，中断运算处理。（定时器的时钟将继续。）
　（b）瞬间掉电解除时，继续运算处理。
　（c）即使发生瞬间掉电并中断运算，警戒定时器（WDT）的计测仍将继续。例如，PC参数的WDT设置为200 ms的情况下，扫描时间为190 ms时发生15 ms的瞬间掉电，就会出现警戒定时器出错。

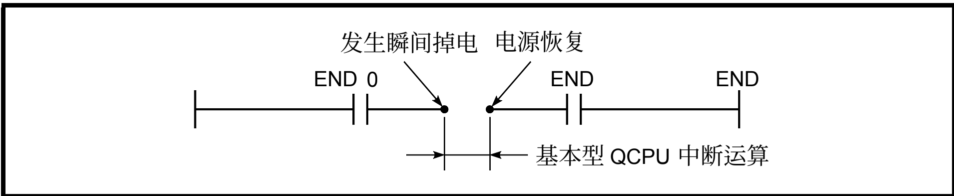


图4.5 发生瞬间掉电时的运算处理

（2）发生超过允许瞬间掉电时间的瞬间掉电时
基本型QCPU进入初始化启动状态（相当于可编程控制器的电源打开时）。进行和以下操作时同样的运算处理。
・打开电源
・利用运行/停止/复位开关进行复位操作
・利用GX Developer等进行远程复位操作

4.6 数据的清零处理

（1）数据的清零

如果基本型QCPU利用运行/停止/复位开关或远程复位进行复位操作，或通过可编程控制器电源的通→断→通进行电源的复位等，除以下数据以外的数据将被清零（位软元件：断，字软元件：0）。

（a）程序存储器的数据
（b）指定为锁存的软元件的数据（锁存清零有效）
（c）指定为锁存的软元件的数据（锁存清零无效）
（d）文件寄存器的数据
（e）故障记录数据（特殊寄存器（SD）存储时）

（b）的数据可以利用来自ＧＸ Developer的远程锁存清零功能进行清零。
远程锁存清零的有关内容请参照7.6.4项。

（2）软元件的锁存指定

（a）软元件的锁存指定（锁存范围的设置）在PC参数的元件设置时按软元件进行设置。
锁存范围的设置有以下2种。
① 锁存清零有效
锁存（1）（可以利用锁存清零功能进行清零）中设置范围的软元件。
设置利用远程锁存清零功能进行锁存清零操作时可以清零的锁存范围。
② 锁存清零无效
锁存（2）（不可利用锁存清零功能进行清零）中设置范围的软元件。
设置利用远程锁存清零功能进行锁存清零操作时不可清零的锁存范围。

（b）设置为锁存清零无效的软元件仅在接到指令，或来自GX Developer的清零操作时方可清零。
① 采用指令的清零方法
采用RST指令复位，或采用MOV/FMOV指令传送“0”。
② 采用GX Developer的清零方法
进行在线的PC存储器清零的软元件寄存器总清（包括锁存）。
GX Developer的操作方法的有关内容请参照GX DEVELOPER的手册。

要　点
文件寄存器清零请采用RST指令复位，或采用MOV/FMOV指令传送K0。

备　注

MOV/FMOV指令的有关内容请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

4.7 输入输出处理和响应延迟

基本型QCPU的输入输出处理采用刷新方式，在结束处理时一次性地进行与输入输出模块之间的通信。

但是，也可以在顺控程序中采用直接存取的输入输出，在顺控程序的各条指令执行时与输入输出模块进行通信的直接方式进行输入输出处理。

直接输入的有关内容请参照10.2.1项；直接输出的有关内容请参照10.2.2项。

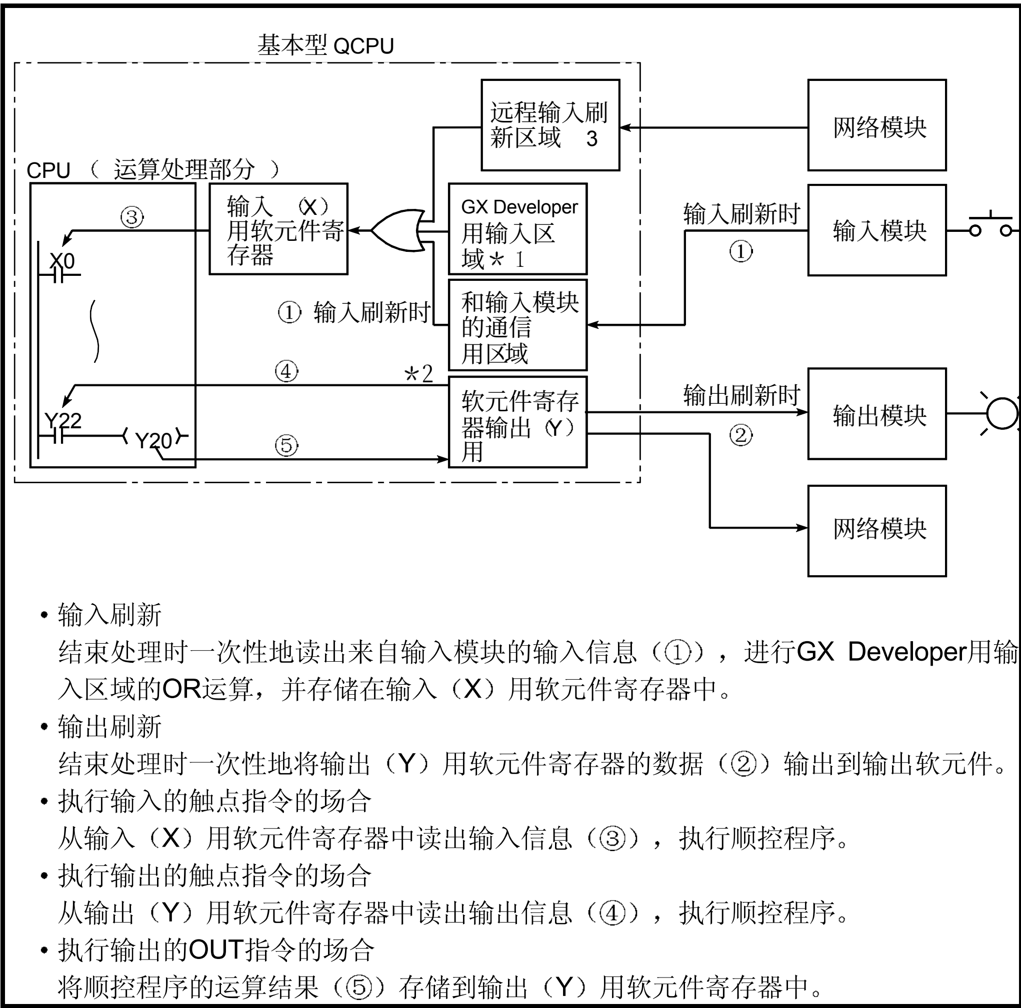
4.7.1 刷新方式

（1）什么是刷新方式

刷新方式就是在结束处理时一次性成批地进行与输入输出模块之间的通信的方式。

（a）输入模块的通/断信息在结束处理时一次性成批地接收到基本型QCPU内部的输入用软元件寄存器中，顺控程序执行时，利用输入用软软元件寄存器的通/断数据进行运算。

（b）输出（Y）的顺控程序的运算结果随时输出到基本型QCPU内部的输出用软元件寄存器中，在结束处理时将输出用软元件寄存器的通/断数据统一输出到输出模块。



- 输入刷新
  结束处理时一次性地读出来自输入模块的输入信息（①），进行GX Developer用输入区域的OR运算，并存储在输入（X）用软元件寄存器中。
- 输出刷新
  结束处理时一次性地将输出（Y）用软元件寄存器的数据（②）输出到输出软元件。
- 执行输入的触点指令的场合
  从输入（X）用软元件寄存器中读出输入信息（③），执行顺控程序。
- 执行输出的触点指令的场合
  从输出（Y）用软元件寄存器中读出输出信息（④），执行顺控程序。
- 执行输出的OUT指令的场合
  将顺控程序的运算结果（⑤）存储到输出（Y）用软元件寄存器中。

图4.6 刷新方式的输入/输出信息的流程

备　注

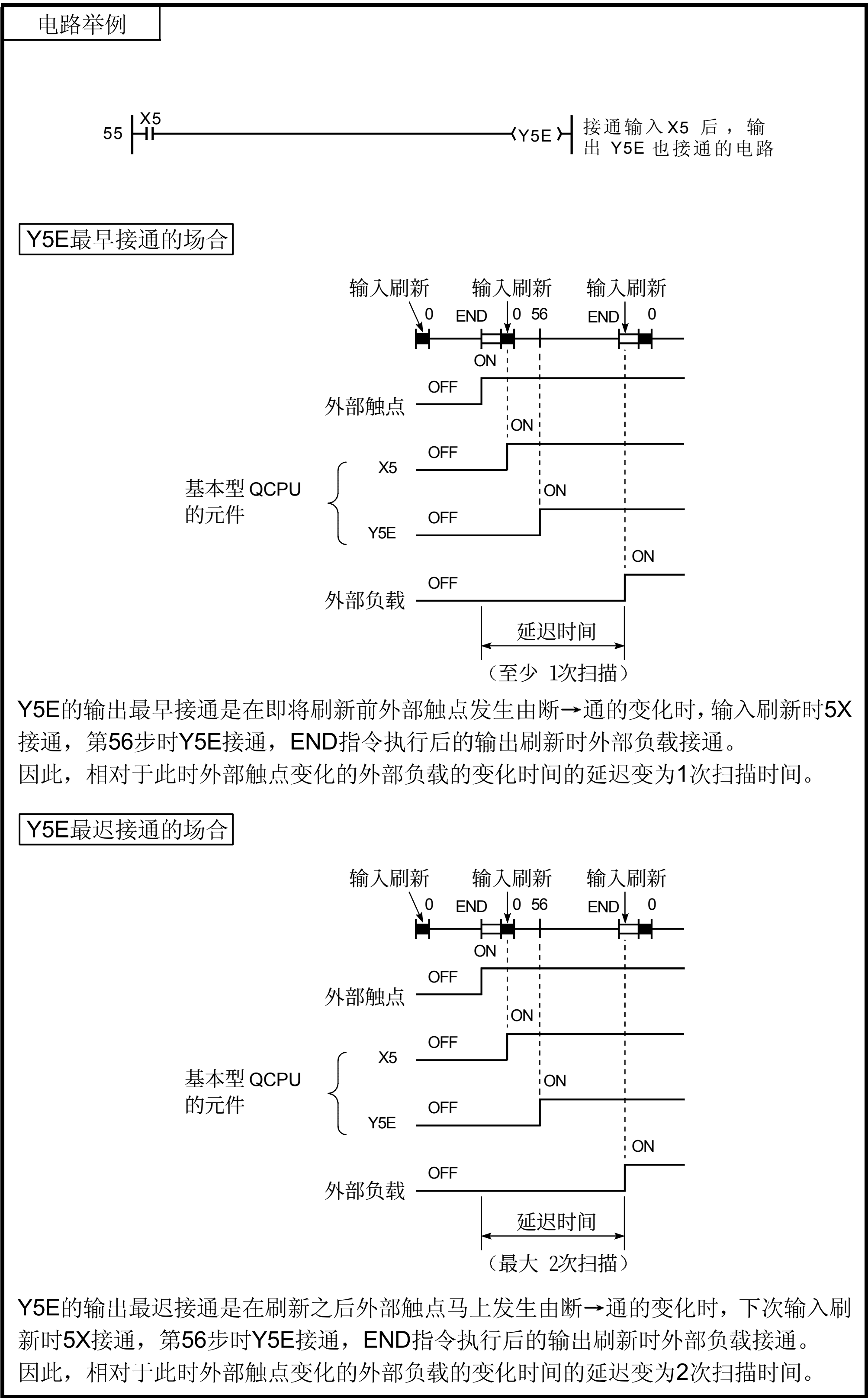
＊1：可以通/断GX Developer用输入区域的有：
- GX Developer的测试操作
- 来自串行通信模块的写入等。

＊2：可以通/断软元件寄存器输出（Y）用的有：
- GX Developer的测试操作；
- MELSECNET/H网络的网络刷新；
- 来自串行通信模块的写入；
- CC—Link的自动刷新。

＊3：远程输入刷新区域表示的是MELSECNET/H、CC—Link中输入（X）设置为自动刷新时的区域。
远程输入刷新区域的自动刷新在结束处理时进行。

（2）响应延迟

相对于输入模块变化的输出变化最大延迟2次扫描时间。（参照图4.8）



Y5E的输出最早接通是在即将刷新前外部触点发生由断→通的变化时，输入刷新时5X接通，第56步时Y5E接通，END指令执行后的输出刷新时外部负载接通。
因此，相对于此时外部触点变化的外部负载的变化时间的延迟变为1次扫描时间。

Y5E的输出最迟接通是在刷新之后外部触点马上发生由断→通的变化时，下次输入刷新时5X接通，第56步时Y5E接通，END指令执行后的输出刷新时外部负载接通。
因此，相对于此时外部触点变化的外部负载的变化时间的延迟变为2次扫描时间。

图4.7 相对于输入X变化的输出Y的变化

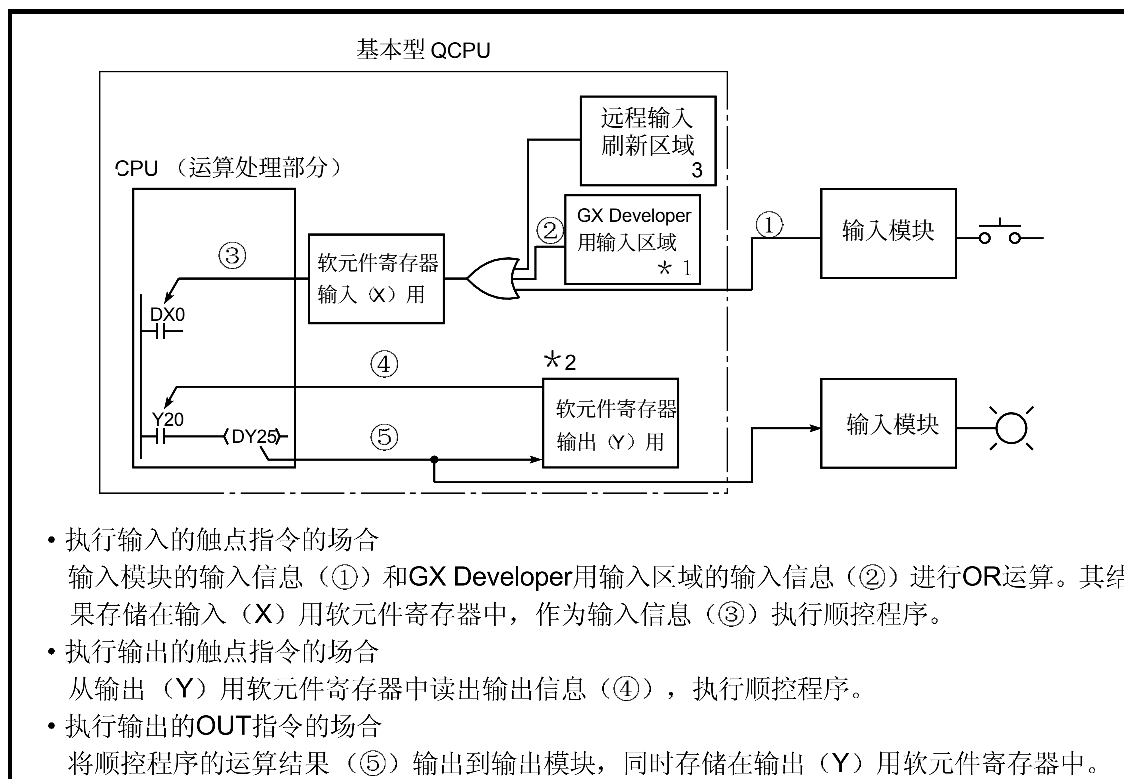
4.7.2 直接方式

（1）什么是直接方式

直接方式就是在顺控程序的各条指令的执行时和输入输出模块进行通信的方式。

在基本型QCPU 上，通过采用直接存取输入（DX）、直接存取输出（Y），可以进行直接方式下的输入输出处理。

直接存取输入的有关内容可参照10.2.1项；直接存取输出的有关内容可参照10.2.2项。



• 执行输入的触点指令的场合
  输入模块的输入信息（①）和GX Developer用输入区域的输入信息（②）进行OR运算。其结果存储在输入（X）用软元件寄存器中，作为输入信息（③）执行顺控程序。
• 执行输出的触点指令的场合
  从输出（Y）用软元件寄存器中读出输出信息（④），执行顺控程序。
• 执行输出的OUT指令的场合
  将顺控程序的运算结果（⑤）输出到输出模块，同时存储在输出（Y）用软元件寄存器中。

图4.8 直接方式的输入/输出信息的流程

备 注

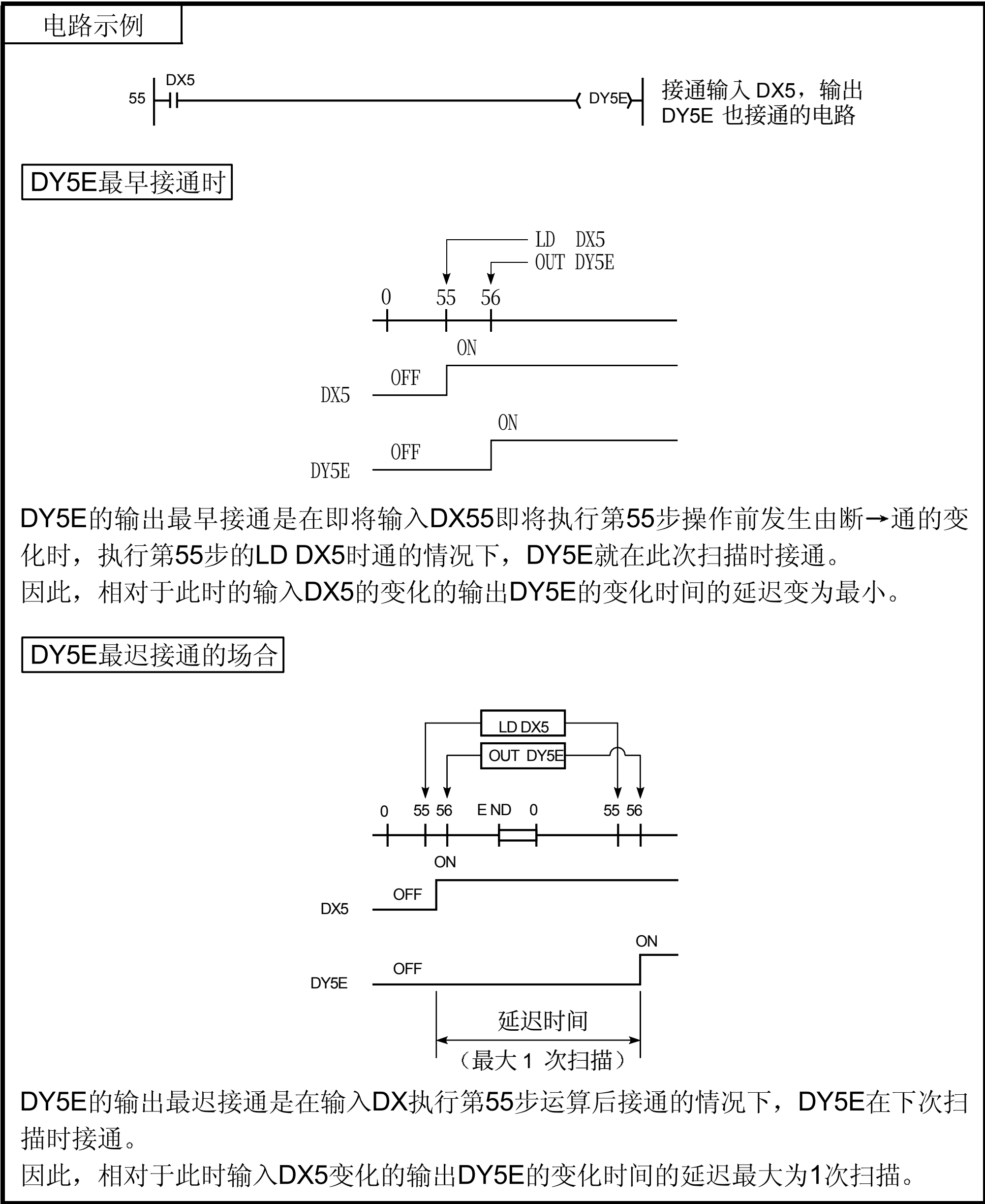
＊1：可以通/断GX Developer用输入区域的有：
  • GX Developer的测试操作；
  • 来自串行通信模块的写入等。

＊2：可以通/断软元件寄存器输出（Y）用的有：
  • GX Developer的测试操作；
  • MELSECNET/H网络的网络刷新；
  • 来自串行通信模块的写入；
  • CC—Link的自动刷新等。

＊3：远程输入刷新区域表示的是MELSECNET/H、CC—Link中输入（X）设置为自动刷新时的区域。
  远程输入刷新区域的自动刷新在结束处理时进行。

（2）响应延迟

相对于输入模块变化的输出变化最大延迟1次扫描时间。（参照图4.10）



DY5E的输出最早接通是在即将输入DX55即将执行第55步操作前发生由断→通的变化时，执行第55步的LD DX5时通的情况下，DY5E就在此次扫描时接通。
因此，相对于此时的输入DX5的变化的输出DY5E的变化时间的延迟变为最小。

DY5E的输出最迟接通是在输入DX执行第55步运算后接通的情况下，DY5E在下次扫描时接通。
因此，相对于此时输入DX5变化的输出DY5E的变化时间的延迟最大为1次扫描。

图4.9 相对于输入X变化的输出Y的变化

4.8 顺控程序中可以使用的数值

基本型QCPU采用0（断）和1（通）的状态来表达数值、字母等数据。
这种用0和1表达的数据称为二进制数（BIN）。
基本型QCPU也可以采用将二进制数按每4位为一组表达的十六进制数（HEX）和二-十进制数（BCD）。
BIN、HEX、BCD及十进制数的数值表达如表4.1所示。

表4.1 BIN、HEX、BCD及二进制数的数值表达

DEC（十进制数）	HEX（十六进制数）	BIN（二进制数）	BCD（二-十进制数）
0	0	0	0
1	1	1	1
2	2	10	10
3	3	11	11
•	•	•	•
•	•	•	•
•	•	•	•
9	9	1001	1001
10	A	1010	1 0000
11	B	1011	1 0001
12	C	1100	1 0010
13	D	1101	1 0011
14	E	1110	1 0100
15	F	1111	1 0101
16	10	1 0000	1 0110
17	11	1 0001	1 0111
•	•	•	•
•	•	•	•
•	•	•	•
47	2F	10 1111	100 0111
•	•		
•	•		
•	•		
32766	7FFE	0111 1111 1111 1110	—
32767	7FFF	0111 1111 1111 1111	—
-32768	8000	1000 0000 0000 0000	1000 0000 0000 0000
-32767	8001	1000 0000 0000 0001	1000 0000 0000 0001
•	•		
•	•		
•	•		
-2	FFFE	1111 1111 1111 1110	—
-1	FFFF	1111 1111 1111 1111	—

（1）从外部向基本型QCPU的数值输入

用数字开关从外部向基本型QCPU设置数值时，可以用BCD（二-十进制数）进行和十进数数同样的设置。

但是，由于基本型采用BIN进行运算，因此，如果将采用BCD码设置的数值照原样使用，就会用不同于设置值的数值进行运算。

为了将采用BCD码设置的数值变换为基本型QCPU所使用的BIN数值，需要用到“BIN指令”。

如果在顺控程序中事先编写好将数值数据变换为BIN的程序，那么，从外部设置数值数据时，就可以不考虑是否是BIN数据。

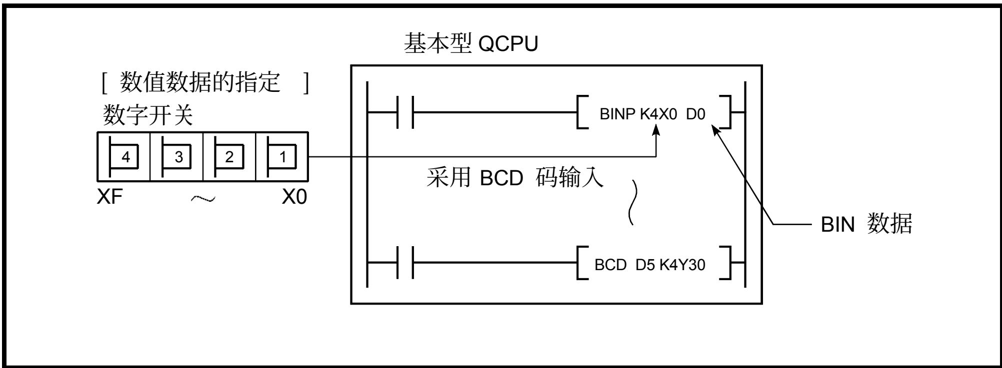


图4.10　采用数字开关向基本型输入数据

（2）从基本型QCPU向外部的数值输出

需要将基本型QCPU上运算的数值在外部显示时，可以使用数字显示器。

基本型QCPU用于运算的BIN数据照原样输出到数字显示器上，也不能正确地显示。

为了将采用BIN进行运算的数据变换为BCD码，需要用到“BCD指令”。

如果在顺控程序中事先编写好将数值数据变换为BCD的程序，那么，就可以和十进制数同样在外部显示了。

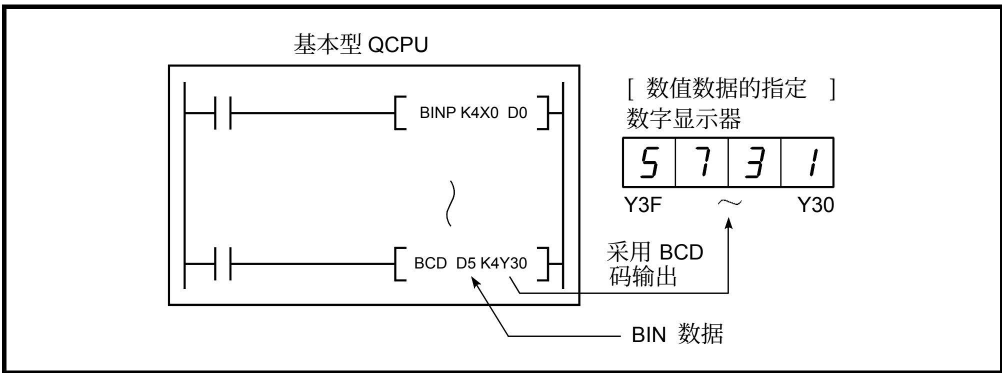


图4.11 采用数字显示器显示基本型QCPU的运算数据

4.8.1 BIN（二进制数：Binary Code）

（1）二进制数

BIN是采用0（断）和1（通）表达的数值。
十进制数从0开始加数到9后，接下去就产生进位，成为10。
BIN在0、1之后就产生进位，成为10（十进制数的2）。
BIN和十进制数的数值表达如表4.2所示。

表4.2　二进制数和十进制数的数值表达的差异

DEC（十进制数）	BIN（二进制数）	
0	0000	
1	0001	进位
2	0010	
3	0011	进位
4	0100	
5	0101	
6	0110	
7	0111	进位
8	1000	
9	1001	
10	1010	
11	1011	

（2）BIN的数值表达

（a）各寄存器（数据寄存器、通信寄存器等）由16位构成。
各寄存器的各个位分配为2^n的数值。
但是，最高位因用作正负判别，不能使用无符号的BIN。
① 最高位为0・・・・・・・・・・正
② 最高为为1・・・・・・・・・・负
各寄存器的数值表达如图4.13所示。

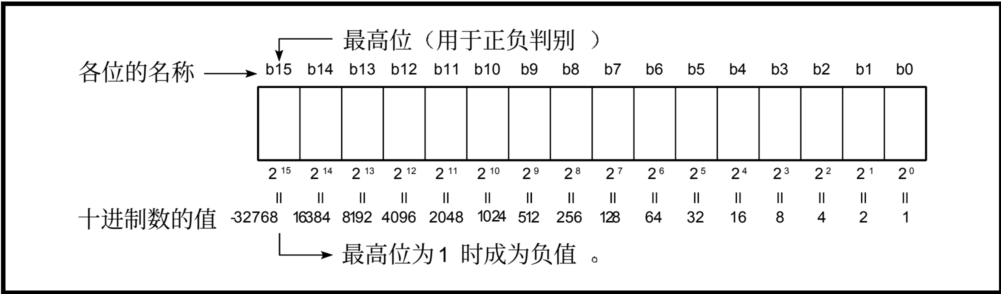


图4.12 各寄存器的数值表达

（b）可以使用的数值数据
采用图4.13所示的数值表达方法可以表达—32768～32767范围内的数值。
因此，各寄存器中可以存储从—32768～32767的数值。

4.8.2 HEX（十六进制数：HEX Decimal）

（1）HEX

HEX是将4位BIN数据作为1位表达的方法。

由于BIN采用0～15表达1位，因此，9之后的10采用字母A表达，11采用字母B表达，F（15）之后产生进位。

BIN、HEX、十进制数的数值表达如表4.3所示。

表4.3 BIN、HEX、十进制数的数值表达

DEC（十进制数）	HEX（十六进制数）	BIN（二进制数）
0	0	0
1	1	1
2	2	10
3	3	11
•	•	•
•	•	•
•	•	•
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	1 0000
17	11	1 0001
•	•	•
•	•	•
•	•	•
47	2F	10 1111

进位

（2）HEX的数值表达

各寄存器（数据寄存器、通信寄存器等）由16位构成。

因此，各寄存器中可以存储的数值用HEX表达时的范围为0～FFFFH。

4.8.3 BCD（二-十进制数：Binary Coded Decimal）

（1）BCD

BCD采用二进制数的表达，但附加有类同于十进制数的进位。
BCD和HEX一样，采用4位表达，但不使用HEX的A～F。
BIN、BCD、十进制数的数值表达如表4.4所示。

表4.4 BIN、BCD、十进制数的数值表达

（十进制数）	BIN（二进制数）	BCD（二-十进制数）
0	0	0
1	1	1
2	10	10
3	11	11
4	100	100
5	101	101
6	110	110
7	111	111
8	1000	1000
9	1001	1001
10	1010	1 0000
11	1011	1 0001
12	1100	1 0010

进位

（2）BCD的数值表达

各寄存器（数据寄存器、通信寄存器等）由16位构成。
因此，各寄存器中可以存储的数值用BCD表达的范围为0～9999。

4.9 字符串数据

（1）字符串

可以处理的是JIS8编码的字符串。

（2）JISS编码字符串

JIS8编码字符串如下表所示。

在下表中，00H（空编码）用作字符串的结束字符。

				b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b4	b3	b2	b1	行＼列	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	NUL		（SP）	0	@	P	`	p				—	タ	ミ		
0	0	0	1	1			!	1	A	Q	a	q			。	ア	チ	ム		
0	0	1	0	2			"	2	B	R	b	r			□	イ	ツ	メ		
0	0	1	1	3			#	3	C	S	c	s			□	ウ	テ	モ		
0	1	0	0	4			$	4	D	T	d	t			、	エ	ト	ヤ		
0	1	0	0	5			%	5	E	U	e	u			.	オ	ナ	ユ		
0	1	1	0	6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
0	1	1	1	7			'	7	G	W	g	w			*ア	キ	ヌ	ラ		
1	0	0	0	8			（	8	H	X	h	x			*イ	ク	ネ	リ		
1	0	0	1	9			）	9	I	Y	i	y			*ウ	ケ	ノ	ル		
1	0	1	0	A			*	:	J	Z	j	z			*エ	コ	ハ	レ		
1	0	1	1	B			+	;	K	[	k	{			*オ	サ	ヒ	ロ		
1	1	0	0	C			(逗号) ,	<	L	¥	l	|			*ヤ	シ	フ	ワ		
1	1	0	1	D			(负号) -	=	M	]	m	}			*ユ	ス	ヘ	ン		
1	1	1	0	E			(句号) 。	>	N	^	n	—			*ヨ	セ	ホ	゛		
1	1	1	1	F			/	?	O	下划线 _	o				*ツ	ソ	マ	。		

* 表示小写字母。

要　　点
字符串只可在字符串传送指令（$MOV）中使用。

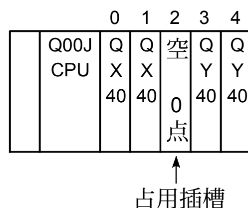
第 5 章　输入输出编号的分配

本章将阐述和输入输出模块、智能功能模块进行数据收发的输入输出编号的分配所必需的有关内容。

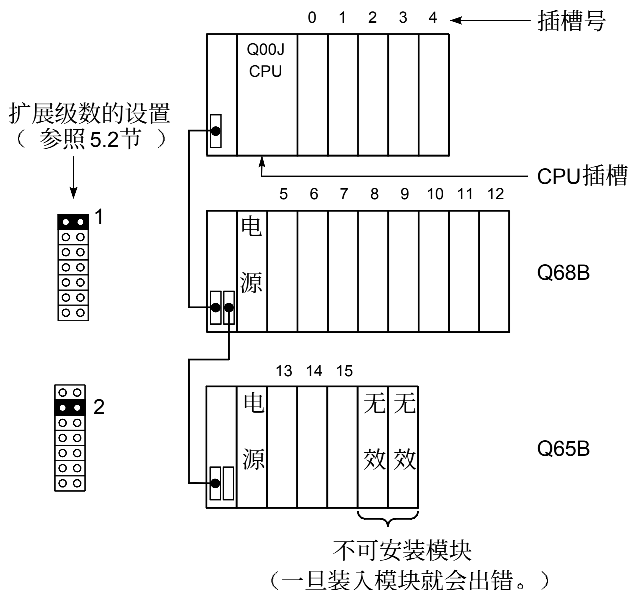
5.1 扩展基板的级数和插槽数的关系

5.1.1 Q00JCPU的场合

Q00JCPU可以由1块主基板和2块扩展基板，合计3块基板来构建系统。
但是，可以使用的插槽数（模块数）为16个插槽，其中包括空插槽。
如下图，即使将插槽2设置为“空0点”，也要占用1个插槽。
因此，下图的系统中就要用到插槽0～插槽4，共5个插槽。



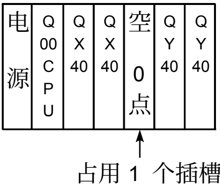
模块请安装在插槽0～插槽15中。
16以后的插槽中一旦装入模块，就会出错（SP. UNIT LAY ERR.）。



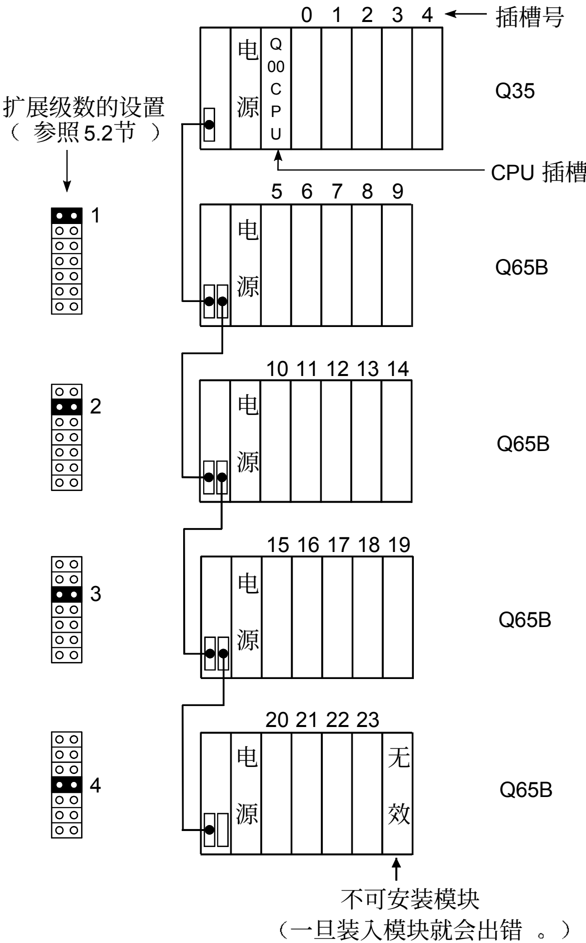
GOT连接到总线上的情况下，须使用扩展1级的1个插槽。而且，1台GOT须占用16个输入输出点。
使用GOT的情况下，请考虑插槽数和输入输出点数。
GOT连接到总线上时的详细内容请参照GOT的手册。

5.1.2 Q00/Q01CPU的场合

Q00/Q01CPU可以由1块主基板和4块扩展基板，合计5块基板来构建系统。
但是，可以使用的插槽数（模块数）为24个插槽，其中包括空插槽。



模块请装入插槽0～插槽23内。
24以后的插槽中一旦装入模块，就会出错（SP. UNIT LAY ERR.）。
因此，下图的系统中就要用到插槽0～插槽4，共5个插槽。



GOT连接到总线上的情况下，须使用扩展1级的1个插槽。
而且，1台GOT须占用16个输入输出点。
使用GOT的情况下，请考虑插槽数和输入输出点数。
GOT连接到总线上时的详细内容请参照GOT的手册。

5.2 扩展基板的安装和级数的设置

扩展基板可以使用装有Q系列对应模块的Q5□B和Q6□B。
扩展基板的QA1S6□B和QA65B则不可使用。

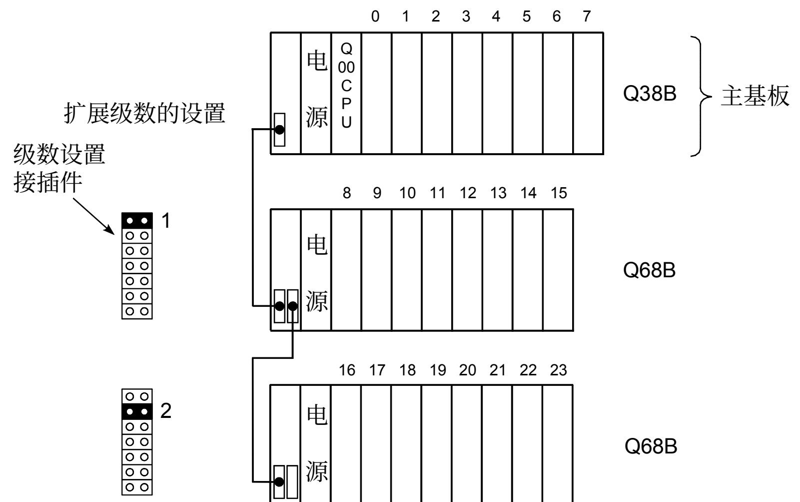
（1）扩展基板的扩展级数的设置顺序
扩展基板必须用级数设置接插件进行扩展级数的设置。
扩展级数请按照主基板上连接的扩展基板的连接顺序设置为1～2/4。

（2）扩展基板的扩展级数设置时的注意事项

（a）扩展级数请连续设置。
基板的分配处于“自动模式”的情况下，即使跳跃式地设置扩展级数，跳过的级数其插槽数仍为0，空的插槽数不会增加。
而且，跳过的级数的输入输出点数也作为0点进行输入输出编号的分配。

（b）多个扩展基板上不可设置同一扩展级数使用。

（c）不可在扩展级数接插件的2处及2处以上插入接插针的状态下使用。
而且，也不可在级数设置接插件上未插入接插针的状态下使用。



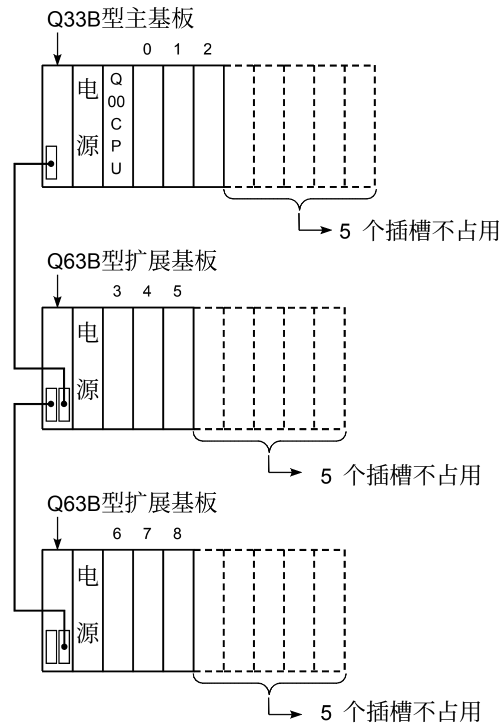
5.3 基板的分配（基本模式）

主基板、扩展基板的模块数的分配有2种模式，即“自动模式”和“详细模式”。

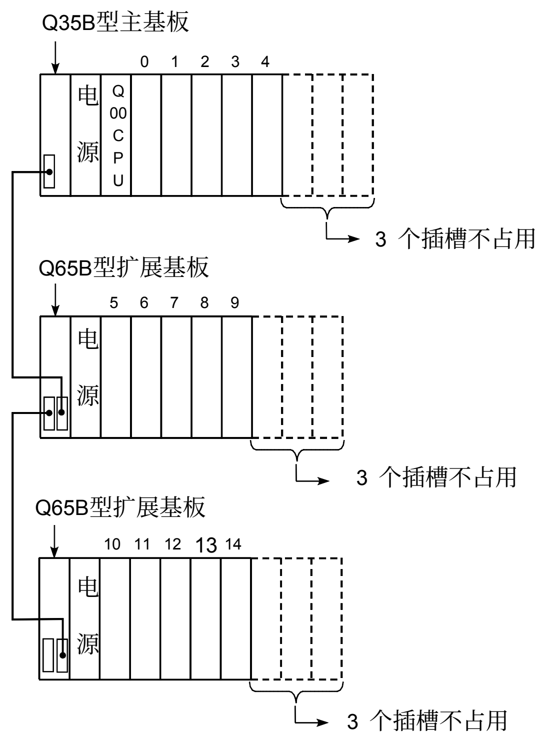
（1）什么是自动模式
自动模式就是利用主基板和扩展基板可以装载的插槽数进行基板分配的模式。
输入输出编号由所使用的基板上装有的模块分配。

AnS系列上主基板和扩展基板固定为8个插槽，因此，即使只使用3个插槽/5个插槽的基板，仍须占用8个插槽。
基本型QCPU只占用基板可以装载的插槽数，因此，使用3个插槽的基板的情况下，就只占用3个插槽。

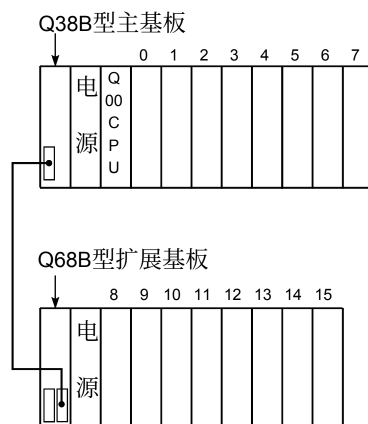
（a）3个插槽的基板的情况下：占用3个插槽



（b）5个插槽的基板/Q00JCPU的场合：占用5个插槽



（c）8个插槽的基板的场合：占用8个插槽



（d）12个插槽的基板的场合：占用12个插槽

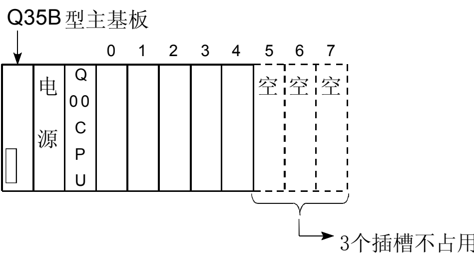


（2）什么是详细模式

（a）详细模式就是在PC参数的输入/输出分配时，将可以安装的模块数按基板（主基板、扩展基板）进行设置的模式。
可以比照AnS系列的基板的占用块数（固定为8个插槽）进行设置。
此外，即使将空插槽在输入/输出分配时设置为0点，也要占用1个插槽。因此，也可以用于不识别未装入模块插槽以后的插槽的场合。

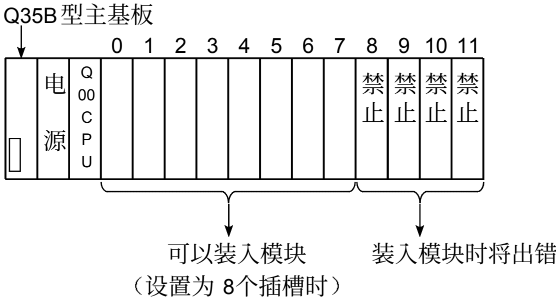
（b）插槽数的设置和注意事项
插槽数的设置可以和所使用的基板的插槽数无关。
但是，正在使用的所有基板必须进行插槽数的设置。
未对所有基板设置插槽数的情况下，输入/输出分配将不能正常动作。
所设置的插槽数和所使用的基板的插槽数不同时，将会出现以下情况。

① 所设置的插槽数多于所使用的插槽数
设置在所使用的基板的插槽数以后的插槽数部分的插槽成为空插槽。
例如，使用5个插槽的基板，而设置为8个插槽时，3个插槽就成为空插槽。



空插槽的点数就是PC参数的PC系统设置时所设置的空插槽点数或PC参数的输入/输出分配时所设置的点数。（默认值为16点。）

② 所设置的插槽数少于所使用的插槽数
将无法使用所设置的插槽数以后的插槽。
例如，使用12个插槽的基板，而设置为8个插槽时，基板上从右面开始的4个插槽将禁止使用。
（禁止使用的插槽中装入模块时将会出错（SP. UNIT LAY ERR.）。）



（3）GX Developer的基本模式的设置画面和设置内容

（a）　（b）　（c）　（d）　（e）

（a）基板型号名称

所安装的基板的型号名称采用16个半角字符进行设置。
基本型QCPU不使用已设置的型号名称。（请作为用户的备望或参数打印时使用。）

（b）电源模块型号名称

所安装的电源模块的型号名称采用16个半角字符进行设置。
基本型QCPU不使用已设置的型号名称。（请作为用户的备望或参数打印时使用。）

（c）扩展电缆型号名称

所使用的扩展电缆的型号名称采用16个半角字符进行设置。
基本型QCPU不使用已设置的型号名称。（请作为用户的备望或参数打印时使用。）

（d）插槽数

所使用的基板的插槽数设置为几点可从以下的插槽数中加以选择。
- 2（2个插槽）
- 3（3个插槽）
- 5（5个插槽）
- 8（8个插槽）
- 10（10个插槽）
- 12（12个插槽）

（e）固定为8块/固定为12块（基本型QCPU上使用）

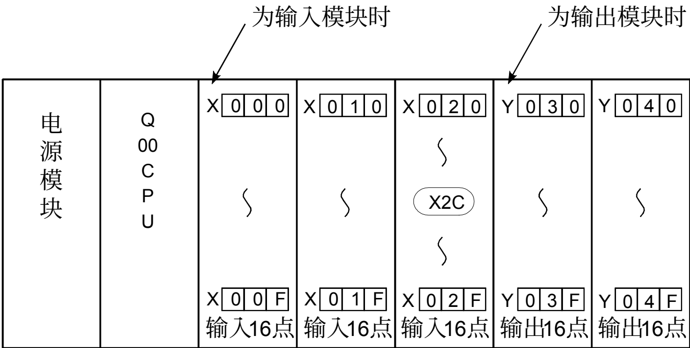
将基板统一设置为指定插槽数时选择。

5.4 什么是输入输出编号

输入输出编号在顺控程序中用于接收通/断数据到基本型QCPU，或从基本型QCPU向外部输出通/断数据。
接收通/断数据到基本型QCPU采用的是输入（X）；从基本型QCPU向外部输出通/断数据采用的是输出（Y）。

输入输出编号采用十六进制数表示。
使用16点的输入输出模块的情况下，输入输出编号如下图所示，每个插槽为从□□0～□□F的16个点，且为连号。

基板上安装的模块上，
- 若是输入模块，则在输出编号的头部附加“X”
- 若是输出模块，则在输出编号的头部附加“Y”



5.5 输入输出编号分配的思路

5.5.1 主基板、扩展基板的输入输出编号

基本型QCPU在电源打开或复位时利用以下所示的项目进行输入输出编号的分配。
因此，即使GX Developer上不进行输入/输出分配，也可以进行基本型QCPU的控制。

分配输入输出编号的情况下，请按以下项目进行输入输出编号的分配。

（1）基板的插槽数
主基板、扩展基板的插槽数根据基本模式的设置变化。（基本模式的有关内容可参照5.3节）

（a）自动模式的情况下，就是基板上可以安装的模块数。
例如，使用5个插槽的基板时就是5个插槽；使用12个插槽的基板时就是12个插槽。

（b）详细模式的情况下，就是PC参数的输入/输出分配时所设置的插槽数。

（2）输入输出编号的分配顺序
输入输出编号以主基板的基本型QCPU的右邻作为0H，向右连续分配编号。

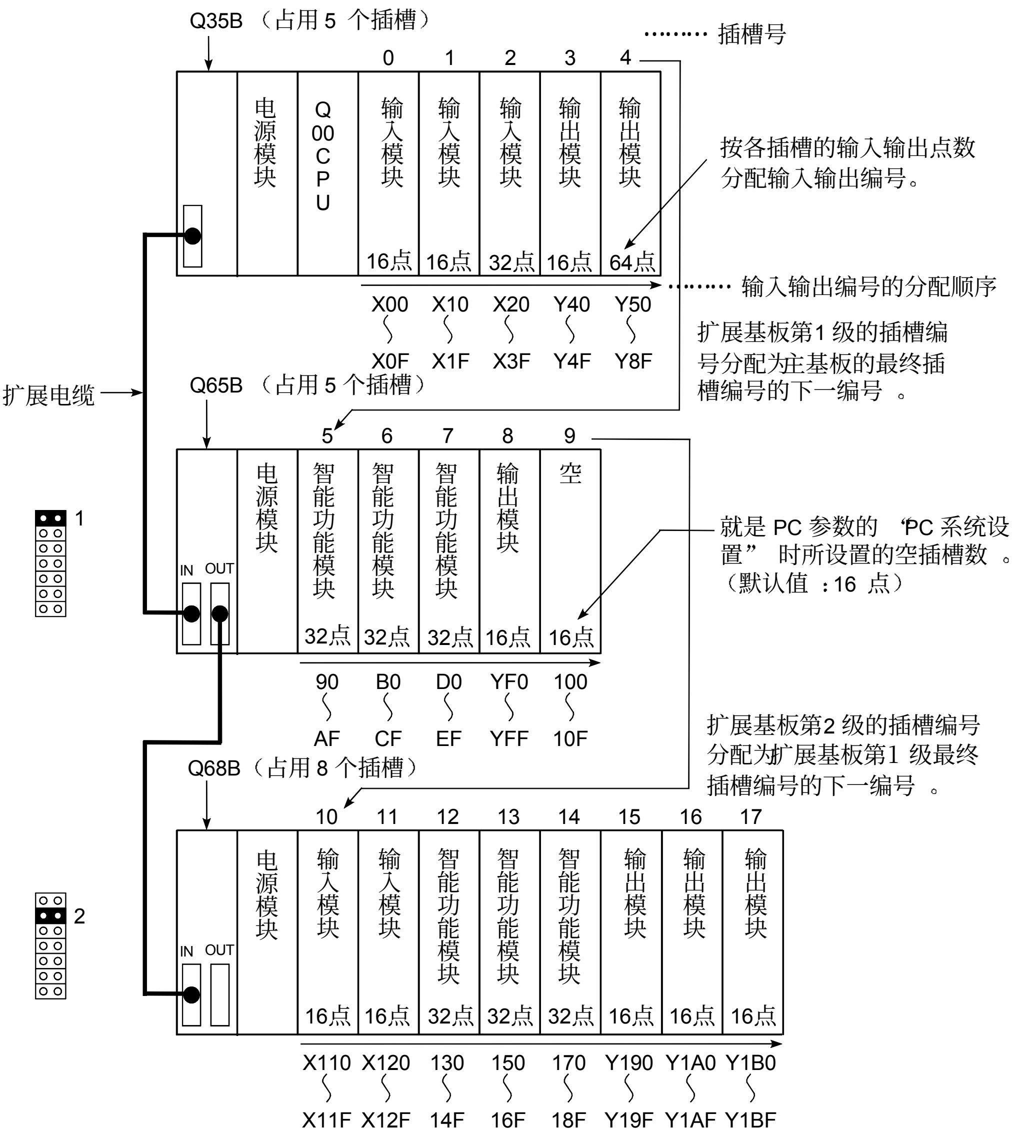
（3）扩展基板的输入输出编号的分配顺序
扩展基板从主基板的输入输出编号的下一个编号开始分配输入输出编号。
扩展基板的分配按照扩展基板的级数设置接插件的设置顺序，从左端（扩展基板的丝印的“I/00”）开始向右连续分配编号。

（4）各插槽的输入输出编号
基板的各插槽占用所安装的输入输出模块、智能功能模块的输入输出点数的输入输出编号。
基本型QCPU的右邻位置如果装有32点的输入模块，则输入输出编号为X0～X1F。

（5）空插槽的输入输出编号
基板上未安装输入输出模块、智能功能模块的空插槽分配为PC参数的PC系统设置时所设置的点数。（默认值为16点。）

要　点
基板的分配采用自动模式进行时，即使采用基板的级数设置接插件跳跃地设置扩展级数，也不能确保空的扩展级数。（输入输出编号向前排。） 如果希望预留空的扩展级数作为将来扩展之用，请利用PC参数对基板进行设置。

基板的设置采用自动模式，不进行输入/输出分配时的输入输出编号的分配举例如下图所示。

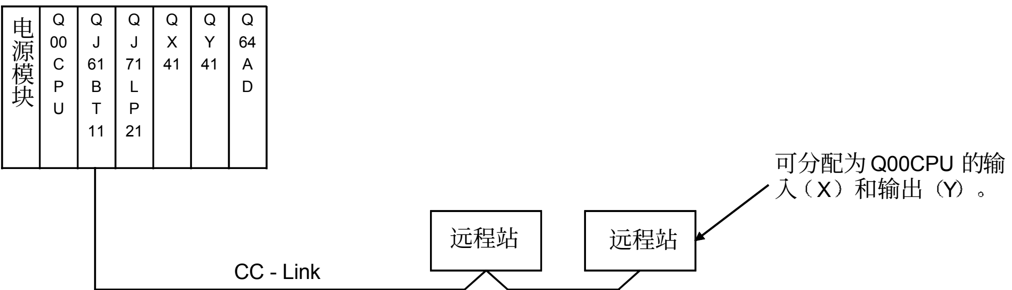


要　点

智能功能模块是为输入输出点数为32点时的情形。
智能功能模块不同，输入输出点数也不同。
请根据所使用的智能功能模块的手册对输入输出点数进行确认后，再分配输入输出编号。

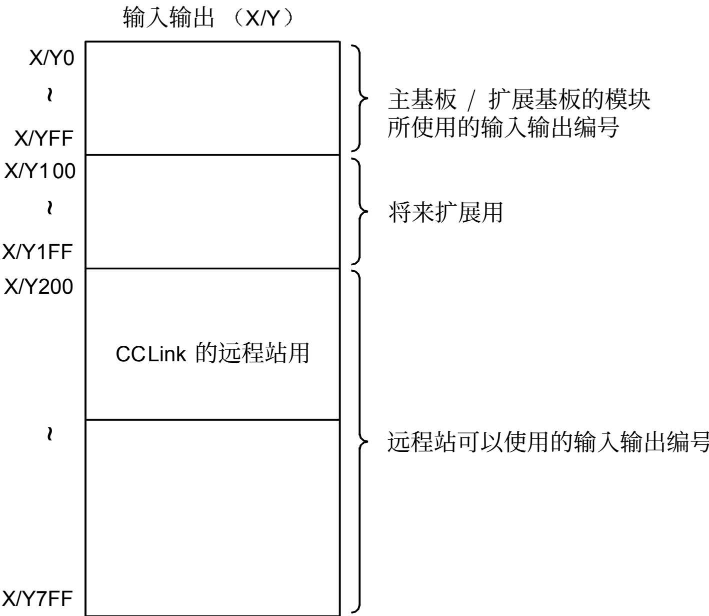
5.5.2 远程站的输入输出编号

在CC—Link的远程输入/输出系统中，可以给远程站的输入输出模块/智能功能模块分配基本型QCPU的软元件的输入（X）和输出（Y）并加以控制。
此外，可以将输入（X）和输出（Y）用作MELSEC/H的通信输入输出（LX、LY）的刷新对象（基本型QCPU一方的软元件）。
MELSEC/H的刷新对象的输入输出编号和CC—Link的远程输入/输出系统的输入输出编号请不要重复。



远程站上使用基本型的软元件的输入（X）和输出（Y）时，分配主基板/扩展基板的输入输出模块/智能功能模块已使用的输入输出编号以后的编号。
例如，主基板/扩展基板的输入输出模块/智能功能模块已使用X/Y0～X/YFF的情况下，可以在远程站上使用X/Y100以后的编号。
但是，请在考虑将来主基板/扩展基板上输入输出模块/智能功能模块扩展的基础上再设置输入输出编号。

（例）主基板/扩展基板上使用X/Y0～X/YFF的256点，且须确保将来扩展所用的X/Y100～X/Y1FF的256点的情况下，就出现下图的结果。



要　　点
CC—Link系统上未进行网络参数设置的情况下，向小输入输出编号的CC—Link的主 ・本地模块分配X/Y400～X/Y7FF的1024个点。

5.6 采用GX Developer器的输入/输出分配

本节阐述采用GX Developer的输入/输出分配的有关内容。

5.6.1 采用GX Developer进行输入/输出分配的目的

采用GX Developer的输入/输出分配在以下情况下进行。

（1）变更为非16点模块时的预留
需要将当前正在使用的模块在将来变更为点数不同的模块时，为了保证在不更改输入输出编号的情况下就能够完成，可以事先预留点数。
例如，将当前安装16点输入模块的插槽分配为32点输入模块。

（2）模块更换时输入输出编号变化的防止
因非16点输入输出模块、智能功能模块的故障而拆除模块时，可以防止输入输出编号的变化。

（3）变更为程序中使用的输入输出编号
所设计的程序中正在使用的输入输出编号和实际的系统输入输出编号不同时，可以将基板的各模块的输入输出编号变更为程序中的输入输出编号。

（4）输入模块、中断模块的输入响应时间的设置（输入/输出响应时间）
根据系统设置输入模块、中断模块的输入响应时间时，在输入/输出分配的 “类别” 选择后再进行。（详细内容请参照7.7节。）

（5）智能功能模块的开关设置
智能功能模块的开关设置在输入/输出分配的 “类别” 选择后再进行。（详细内容请参照7.8节。）

（6）基本型QCPU出错时的输出设置
基本型QCPU发生停止出错，停止基本型QCPU的运算时的输出模块和智能功能模块的输出（保持/清零）的设置在输入/输出分配的 “类别” 选择后再进行。

（7）智能功能模块的硬件出错时的基本型QCPU的动作设置
智能功能模块上发生硬件出错时的基本型QCPU的动作（继续/停止）状态的设置在输入/输出分配的 “类别” 选择后再进行。

要　点
（1）PC参数的输入/输出分配设置在可编程控制器电源接通或基本型QCPU复位时有效。 更改PC参数时，请重新打开可编程控制器的电源或进行基本型QCPU的复位。 （2）输入模块的响应时间的设置、智能功能模块的开关设置时，必须进行输入/输出分配。

5.6.2 采用GX Developer进行输入/输出分配的思路

（1）按插槽进行输入/输出分配

基板的各插槽可以单独设置“类别”（模块类别）、“点数”（输入输出点数）和“起始XY”（首I/O编号）。

例如，变更指定插槽的输入输出点数时，可以只设置点数。

不设置的项目就采用基板的安装状态。

输入/输出分配在PC参数的输入/输出分配时进行。

（a）（b）（c）（d）（e）

（a）插槽

表示插槽号和基板的第几级的第几个插槽。

基板不设置为详细模式的情况下，基板的第几级变为“*”，第几个插槽变为主基板的第0个插槽开始的插槽数。

（b）类别

从以下项目中选择所安装的模块的类别。

- 空（空插槽）
- 输入（输入模块）
- 高速输入（Q系列对应的高速输入模块）
- 输出（输出模块）
- 输入输出混合（输入输出混合模块）
- 智能（智能功能模块）
- 中断（Q系列对应中断模块）

不设置类别的插槽采用实际安装的模块的类别。

（c）型号名称

所安装的模块的型号名称采用16个半角字符进行设置。

基本型QCPU不使用所设置的型号名称。（作为用户的备忘录。）

（d）点数（基本型QCPU上使用）

变更各插槽的输入输出点数时，从以下的点数中加以选择。

- 0（0点）
- 16（16点）
- 32（32点）
- 48（48点）
- 64（64点）
- 128（128点）
- 256（256点）
- 512（512点）＊
- 1024（1024点）＊

不设置点数的插槽采用实际安装的模块的点数。

＊：仅在Q00/Q01CPU上可以设置。

（Q00JCPU的输入输出点数为256点，因此，不可设置为512/1024点。）

（e）起始XY（基本型QCPU上使用）

① 变更各插槽的输入输出点数时，设置变更后的起始输入输出编号。
不设置起始XY的插槽根据已设置的插槽采用连号分配输入输出编号。

② 各插槽的输入输出编号的设置请不要和基本型QCPU所分配的输入输出编号重复。
输入输出编号出现重复时，将会出错（SP. UNIT LAY ERR.）。

（2）进行过输入/输出分配的插槽的状态

进行过输入/输出分配的插槽以输入/输出为优先，与模块的实际安装无关。

（a）设置的点数少于实际安装的输入输出模块数时，实际安装的输入输出模块的实际使用数将会减少。
例如，安装32点输入模块的插槽在输入/输出分配时设置为16点的输入模块时，32点输入模块的后半部分的16点将无法使用。

（b）设置的点数多于实际安装的输入输出模块数时，超过实际安装数的部分将失效。

（c）实际安装的模块和输入/输出分配的类别请设置为同一类别。
输入/输出分配和实际安装的模块的类别不同时，不能正常动作。

实际安装模块	输入/输出分配的设置	结　　果
输入模块	输出/空	空
输出模块	输入/空	空
输入模块/输出模块	智能	出错（SP. UNIT LAY ERR.）
智能功能模块	空	空
	输入/输出	出错（SP. UNIT LAY ERR.）
空插槽	智能	不出错

（d）设置的点数少于实际安装的智能功能模块数时，出现“SP. UNIT LAY ERR.”。

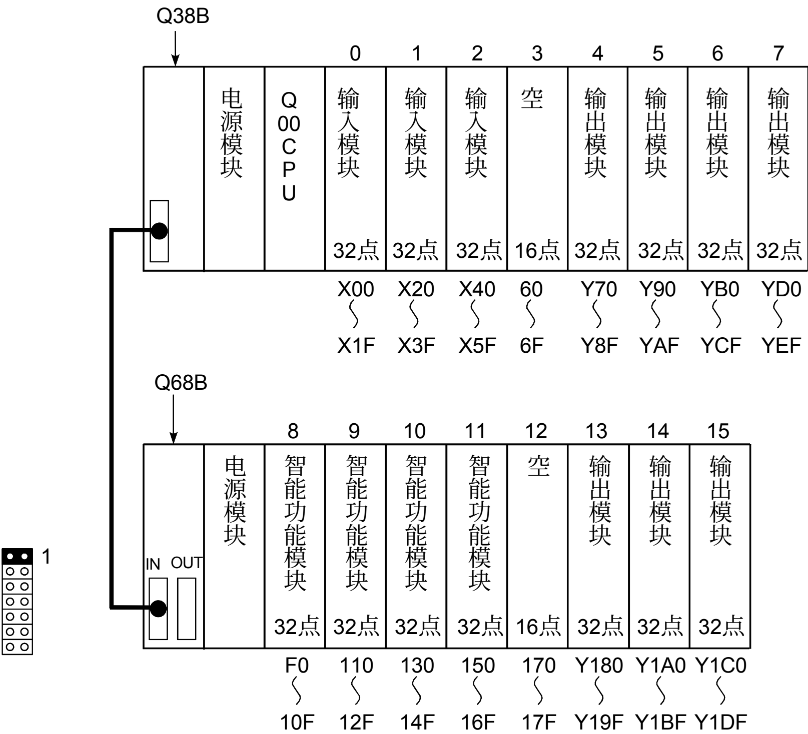
（e）进行输入/输出分配时，请将最终的输入输出编号设置在FFH/3FFH的范围内。
最终输入输出编号的设置超过FFH/3FFH时，将会出错（SP. UNIT LAY ERR.）。
（GX Developer的系统监视将输入/输出地址显示为＊＊＊。）

5.7 输入输出编号的分配举例

本节介绍采用GX Developer进行输入/输出分配时的输入输出编号分配的实例。

（1）空插槽的点数由16点变为32点
空插槽的位置（3号插槽）处将来要安装32点输入模块时，为使输入输出编号不发生变化，进行32点预留。（12号插槽的空插槽为16点，不进行变更。）＊1

（a）系统构成和输入/输出分配前的输入输出编号的分配



备 注

＊1：PC参数的PC系统设置的空插槽数为16点时的情形。
＊2：Q00CPU的输入输出点数为256，因此，请在X/Y0～X/YFF的范围内使用。

（b）采用GX Developer的输入/输出分配

在GX Developer的输入/输出分配设置画面下将插槽号3设置为“32点”。

选择32点。
（不选择类别的情况下，采用所安装的模块的类别。）

Qn(H)パラメータ設定

PCネーム設定 | PCシステム設定 | PCファイル設定 | PC RAS設定 | デバイス設定
プログラム設定 | ブートファイル設定 | SFC設定 | I/O割付設定

I/O割付

	スロット	種別	形名	点数	先頭XY
0	CPU	CPU			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				
4	3(*-3)		QX41	32点	
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

先頭XYは未入力の場合PCが自動で割り付けます。
先頭XYが未入力の時はチェックでエラーとならない場合があります。

スイッチ設定　詳細設定

基本設定

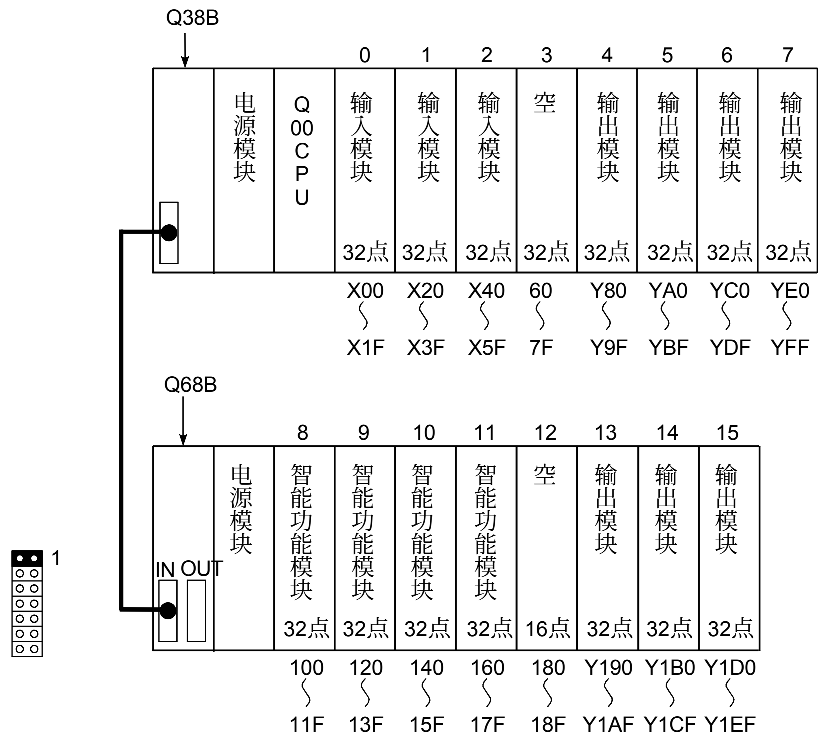
	ベース形名	電源ユニット形名	増設ケーブル形名	スロット数
基本				
増設1				
増設2				
増設3				
増設4				
増設5				

ベースモード：自動　詳細

8枚固定　12枚固定　PCデータ読出

X/Y割付確認　デフォルト　チェック　設定終了　キャンセル

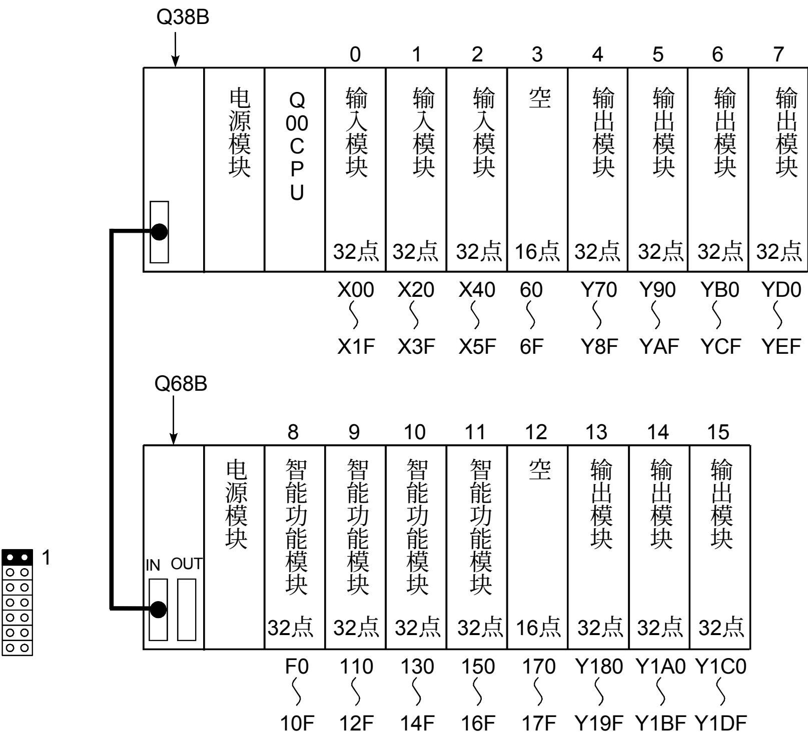
（c）输入/输出分配后输入输出编号的分配



（2）更改插槽的输入输出编号

为了在目前为空的插槽位置（3号插槽）上安装32点输入模块时4号插槽以后的输入输出编号不发生变化，将3号插槽的输入输出编号变更为X200～X21F。＊1

（a）系统构成和输入/输出分配前的输入输出编号的分配

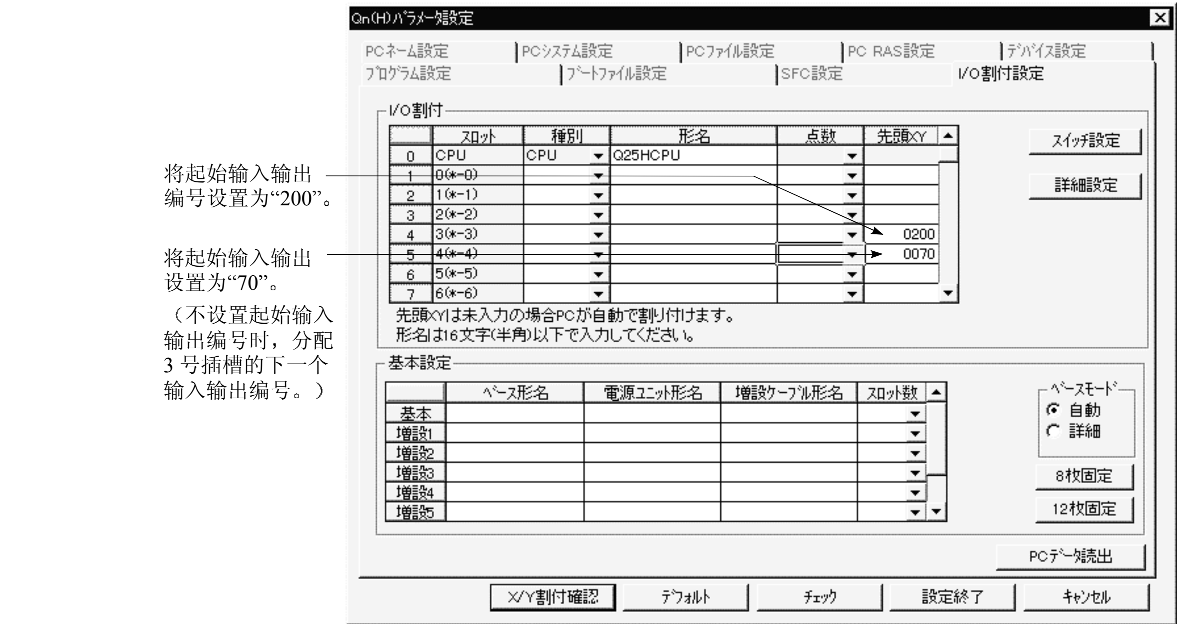


备 注

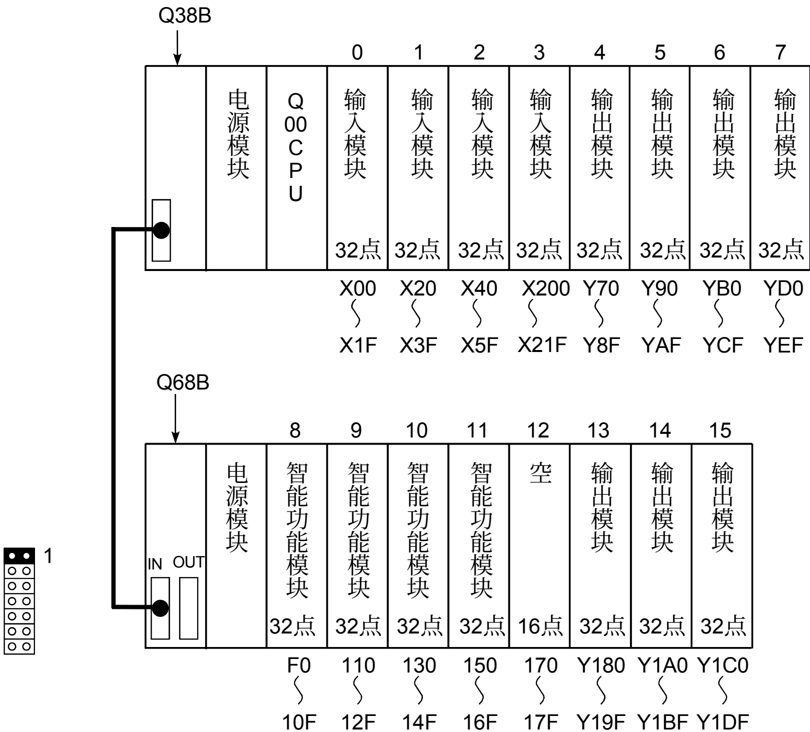
＊1：Q00JCPU的输入输出点数为256点，因此，请在X/Y0～X/YFF的范围内使用。

（b）采用GX Developer的输入/输出分配

在GX Developer的输入/输出分配设置画面下将插槽号3设置为“200”；将插槽号4设置为“70”。



（c）输入/输出分配后输入输出编号的分配



5.8 输入输出编号的确认

可以通过GX Developer的系统监视对基本型QCPU上安装的模块和输入输出编号进行确认。

（系统监视的有关内容请参照7.18节。）

第6章　关于基本型QCPU所处理的文件

（1）基本型QCPU所处理的数据

基本型QCPU将参数・程序・软元件注释等数据存储在程序存储器中。

此外，进行ROM运行时，将程序存储器中的参数・程序写入标准ROM。

（2）采用GX Developer的参数・程序的写入

参数•程序•软元件注释等数据采用GX Developer（在线PC写入）写入基本型QCPU的程序存储器。

在线PC写入时，指定写入基本型QCPU的数据类别（参数・程序・注释等）。

6.1 关于基本型QCPU的存储器

（1）用户存储器

用户存储器就是基本型QCPU的存储器中可以由用户通过GX Developer/顺控程序进行读写的存储器。

用户存储器有以下几种。

- 程序存储器
- 标准ROM

此外，Q00/Q01CPU上还有内置标准RAM。

（a）程序存储器

这是基本型QCPU用于存储进行实际运算的程序的存储器。

存储在标准ROM中的程序引导（读出）到程序存储器中后进行运算。（引导运行）

此外，程序存储器中的参数・程序也可以统一复制到标准ROM中。（程序存储器的ROM化）

（b）标准ROM

这是基本型QCPU上进行ROM运行时存储参数・程序等数据的存储器。

（c）标准RAM

这是存储文件寄存器的数据的存储器。

标准RAM的文件寄存器可以实现和数据寄存器同样的高速存取。

6

（2）基本型QCPU中可存储的数据

基本型QCPU的程序存储器、标准RAM、标准ROM中可以存储的数据如下表所示。

数据名	Q00JCPU内置		Q00/Q01CPU内置			文件名和扩展符
	程序寄存器	标准ROM	程序寄存器	标准ROM	标准ROM	
参数	◎	○	◎	×	○	PARAM.QPA
智能功能模块参数	○	○	○	×	○	IPARAM.QPA
程序	◎	○ *1	◎	×	○ *1	MAIN.QPG
文件寄存器	×	×	×	○ *3	×	MAIN.QDR
软元件注释	○ *2	○ *2	○ *2	×	○ *2	MAIN.QCD

◎：必要数据、○：可存储的数据、×：不可存储的数据

备 注

*1：实际执行程序时，必须用PC参数向程序存储器作引导指定。

*2：可以从GX Developer写入。

但是，不可采用顺控程序的指令来使用软元件注释。

*3：标准RAM中只能将32k点的文件寄存器存储在1个文件中。

（3）驱动器号

（a) 基本型QCPU上采用驱动器号对程序存储器、标准RAM和标准ROM进行管理，但使用GX Developer时可以不必在意驱动器号。
GX Developer指定对象存储器（程序存储器、标准RAM、标准ROM）并对基本型QCPU进行参数、程序等的写入/读出。

（b）从串行通信模块等读出/写入时的对象存储器（程序存储器、标准RAM、标准ROM）的指定采用下表所示的驱动器号进行。

<table>
<tr><th colspan="2">存储器</th><th>驱动器号</th></tr>
<tr><td rowspan="3">基本型QCPU内置</td><td>程序存储器</td><td>0</td></tr>
<tr><td>标准RAM</td><td>3</td></tr>
<tr><td>标准ROM</td><td>4</td></tr>
</table>

（4）存储器容量和格式化的要否
基本型QCPU的各个存储器的存储容量和格式化的要否如下表所示。

<table>
<tr><th></th><th>Q00JCPU</th><th>Q00CPU</th><th>Q01CPU</th><th>格式化的要否</th></tr>
<tr><td>标准RAM（k字节）</td><td>无</td><td colspan="2">64</td><td>要　*</td></tr>
<tr><td>程序存储器（k字节）</td><td>58</td><td>94</td><td>94</td><td>要　*</td></tr>
<tr><td>标准ROM（k字节）</td><td>58</td><td>94</td><td>94</td><td>不要</td></tr>
</table>

*：使用前请务必利用GX Developer进行格式化。
（但是，如果存储器处于初始状态，或因电池用完（Q6BAT）而存储器处于不稳定状态时，可编程控制器的电源接通或复位时将自动进行格式化。）

6.2 关于程序存储器

（1）什么是程序存储器

（a）程序存储器就是RAM存储器（随机存取存储器），用于存储基本型QCPU上执行的程序。

（b）程序存储器的数据由基本型QCPU上安装的电池（Q6BAT）来维持。

（c）使用前请务必采用GX Developer对程序存储器进行格式化。*1
（但是，如果基本型QCPU处于初始状态，或因电池用完（Q6BAT）而存储器处于不稳定状态时，可编程控制器的电源接通或复位时将自动进行格式化。）
采用GX Developer进行格式化时，请参照GX Developer的手册。

表6.1 格式化后的存储器容量

CPU型号名称	存储器容量	可存储程序数
Q00JCPU	58 k字节	1
Q00CPU	94 k字节	1
Q01CPU	94 k字节	1

要　点

程序在程序存储器中以4字节单位存储。

（2）存储的数据
程序存储器中可以存储参数、程序等数据。
关于程序存储器中可以存储的数据，请参照6.1节。

备　注

*1：采用GX Developer对程序存储器进行格式化时，也可以对系统区域中的用户设置区域进行分配。（系统区域中的用户设置区域可以以1 k步为单位，设置0～3 k步。）
系统区域中的用户设置区域用于来自GX Developer的监视数据注册，该GX Developer连接在串行通信模块等上。
事先设置好系统区域中的用户设置区域，就可以尽早从连接到串行通信模块上的GX Developer上进行监视。
但是，一旦设置系统区域中的用户设置区域，作为用户文件区域使用的区域将会减少。

6.3 关于标准ROM

（1）什么是标准ROM

（a）标准ROM就是基本型QCPU上用于ROM运行的存储器。

（b）标准ROM中存储的程序进行PC参数的引导设置，引导（读出）到程序存储器中后开始使用。

（c）标准ROM的写入利用GX Developer的在线“PC写入（闪存ROM）”的“程序存储器的ROM化”进行。（参照6.4.1项）

要　　点
程序存储器一旦进行ROM化，程序存储器中的数据将原封不动地复制到标准ROM中。

（2）存储的数据

标准ROM中可以存储参数、程序等数据。
关于标准ROM中可以存储的数据，请参照6.1节。

（3）ROM运行的设置

进行ROM运行时，须在PC参数的引导文件设置时进行 “从标准ROM进行引导运行” 的设置。

6.4 标准ROM的程序执行（引导运行）和程序存储器的ROM化

6.4.1 标准ROM的程序执行

（1）基本型QCPU的程序执行

（a）基本型QCPU对程序存储器中存储的程序进行运算。
标准ROM中存储的程序不进行运算。

（b）标准ROM中存储的程序先引导（读出）到程序存储器后再进行运算。

（2）引导运行之前的步骤

进行引导运行之前的步骤如下所示。

（a）采用GX Developer编写程序
编写引导运行的程序。

（b）采用GX Developer作引导设置
通过PC参数的引导文件设置，设置为“从标准ROM进行引导”。



（c）采用GX Developer将参数、程序等写入标准ROM

① 利用GX Developer的在线“PC写入”将参数和程序写入程序存储器。

② 将写入程序存储器中的参数和程序通过程序存储器的ROM化，向标准ROM中写入参数程序等的文件。
采用GX Developer向标准ROM写入参数程序等有关内容请参照6.4.2项。

（d）程序的执行
一旦利用运行/停止/复位开关对基本型QCPU进行复位，就从标准ROM进行引导。
基本型QCPU的复位操作的有关内容请参照下列手册。
• QCPU（Q模块）用户手册（硬件篇）
• 基本型QCPU（Q模块）用户手册（硬件设计 • 保养检点篇）

（3）标准ROM的程序执行时的注意事项
（a）进行引导运行时，请将参数（PC参数）和程序存入标准ROM。

（b）从ROM进行引导运行时，即使对程序存储器内运行中的程序进行写入操作，变更内容也不会反映在引导源的标准ROM中的程序上。

（c）向程序存储器写入顺控程序并对可编程控制器的电源进行接通/复位操作时，如果程序存储器中的内容发生改写，可以考虑是进行了引导运行的缘故。
PC参数的引导文件设置中设置为“从标准ROM进行引导运行”时，请进行以下的操作。
① 进行程序存储器的格式化
② 进行程序存储器的ROM化
（标准ROM的参数、顺控程序被消去。）
③ 向程序存储器写入参数、顺控程序。

6.4.2 程序存储器的ROM化

从GX Developer向标准ROM的写入利用GX Developer的在线PC写入(闪存ROM)的"程序存储器的ROM化"进行。
标准ROM不可用GX Developer的在线"PC写入"进行文件的写入操作。

(1) 程序存储器的ROM化

(a) 将程序存储器中存储的文件一次性成批地写入标准ROM。
程序存储器ROM化的步骤请参照12.1节。
用于程序存储器中进行过调试的程序的ROM化。

(b) 程序存储器进行ROM化，就在标准ROM中存储的文件全部消去后，将程序存储器中的文件一次性成批地写入。
因此，标准ROM中存储的文件不可进行追加。

(c) 标准ROM所使用的存储容量相当于程序存储器所使用的存储容量。
不可使用比程序存储器所使用的存储容量更大的存储器。

(d) 从GX Developer进行程序存储器的ROM化时，如果GX Developer的时间检查时间少于180秒，就进行180秒的检查。
从经过CC—Link的其他站的GX Developer进行程序存储器的ROM化时，请将CC—Link的CPU监视时间设置为180秒以上。

6.5 关于标准RAM

（1）什么是标准RAM

（a）标准RAM就是文件寄存器所用的存储器。

（b）标准RAM的数据由CPU模块上安装的电池（Q6BAT）来加以维持。
即使是在将程序写入标准ROM，进行ROM运行的情况下，如果需要将标准RAM用于文件寄存器，就必须用到电池。

（c）使用前请务必利用GX Developer对程序存储器进行格式化。
（但是，如果基本型QCPU处于初始状态，或因电池用完（Q6BAT）而存储器处于不稳定状态时，可编程控制器的电源接通或复位时将自动进行格式化。）
利用GX Developer进行格式化时，请参照GX Developer的手册。

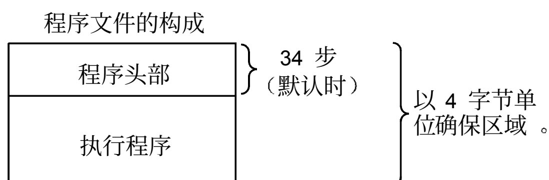
（d）标准RAM的写入采用在线的“PC写入”进行。

（2）存储的数据
标准RAM中可以存储1个文件寄存器的文件。
（标准RAM中不可存储文件寄存器以外的文件。）

6.6 程序文件的构成

（1）程序文件的构成

（a）程序文件由程序头部和执行程序构成。



（b）存储在基本型QCPU的程序存储器中的程序容量为上述2种区域的合计。

① 文件头部

存储文件名、文件尺寸、建立文件的日期等的区域。

文件头部的尺寸为34～35步（136～140字节）。（默认值为34步）

② 执行程序

存储所编写的程序的区域。

每步为4个字节。

（2）GX Developer上对程序容量的确认

GX Developer编程时，程序容量（文件头部的容量和程序步数的合计）如下图以步数的形式显示。

编写程序时，可以确认所编写的程序的容量。



6.7 采用GX Developer的文件操作和文件操作时的注意事项

6.7.1 文件的操作

程序存储器、标准ROM中存储的文件可以通过GX Developer的在线操作进行表6.2中的文件操作。
但是，采用GX Developer的口令注册根据基本型QCPU的运行/停止/复位的状态，可执行的文件操作会有所不同。

表6.2 可以从GX Developer执行的文件操作一览

文件的操作	可否操作			操作内容
	A	B	C	
PC读出	○	△	○	从对象存储器中读出文件。
PC写入	△	△	×	向程序存储器中写入文件。
PC对照	△	△	○	将对象存储器和GX Developer的文件核对。
程序存储器的ROM化	△	△	×	将程序存储器中存储的文件一次性成批地写入标准ROM。
PC数据删除	△	△	×	删除存储器上存储的文件。
PC存储器格式化	○	○	×	对存储器进行格式化。
PC存储器整理	○	○	×	将存储器上的配置不连续的文件重新配置。
电路模式下运行中的写入	△	△	○	将电路模式下变更的内容写程序存储器。

○：可以执行、△：口令一致时可以执行、×：不可执行

备　注

可否操作记号的内容如表6.3所示。

表6.3 可否操作记号的内容一览表

记 号	内　容
A	对文件设置写保护的口令时
B	对文件设置读/写保护的口令时
C	基本型QCPU处于运行状态时

6.7.2 文件操作时的注意事项

（1）文件操作时发生电源切断（包括复位）

（a）不发生文件移动的文件操作正在进行的过程中发生电源切断的情况下，各存储器的数据将不定。

（b）基本型QCPU上利用电池（Q6BAT）进行备份时，如果在发生文件移动的以下操作正在进行的过程中发生电源切断，程序存储器中的数据将不定。

- 文件容量变更
- PC存储器的整理
- 文件的新建

（2）从多个GX Developer同时存取同一文件

在基本型QCPU 上，1个文件只可从1个GX Developer上进行存取。
因此，从多个GX Developer上同时存取同一文件时，请在一个GX Developer的处理完毕后，再进行下一个GX Developer的处理。

6.7.3 文件的存储器容量

根据所使用的文件种类，文件的大小将有所不同。
使用程序存储器、标准RAM、标准ROM的情况下，请利用表6.4大致算出各个文件的大小。
根据所存储的存储区域，采用以下的容量单位确保容量。
• 程序存储器、标准ROM：4个字节

表6.4 文件的存储容量一览表

功　能	大致文件容量（单位：字节）
驱动标题	64
参数	默认值：522（根据参数的设置增加。） 参 考 引导设置→70＋（18×（文件数）） 有MELSECNET/H设置→最多增加（4096） 有Ethernet设置→最多增加（922） 有CC—Link设置→最多增加（254/模块）
顺控程序	136＋（4×（步数））
软元件注释	74＋（各软元件的注释数据尺寸的合计） • 1个软元件的注释数据尺寸＝10＋10250×a＋40×b • a：（（软元件数）/256）的商 • b：（（软元件数）/256）的余数
文件寄存器	2×（文件寄存器数）

以下是程序存储器中写入了参数和顺控程序时的存储容量的计算实例。

（1）写入文件

文件名	程序容量＊
参数	—
顺控程序	5000步（20000字节）

＊：表示GX Developer上显示的程序容量（程序头部和所编写的程序步数的合计）。（参照6.6节）

（2）写入条件

参数：默认设置（522字节）

（3）文件存储容量的计算

文件名	文件容量		存储容量＊
参数	522字节		522字节
顺控程序	顺控程序容量	20，000字节	20000字节
文件的存储容量合计			20384字节

＊：程序存储器以4字节单位（每步）确保存储容量。

第7章　功能

本章阐述基本型QCPU的功能方面的有关内容。

7.1　功能一览

基本型QCPU的功能一览表如下所示。

项　　目	内　　容	参　照
恒定扫描	保持扫描时间恒定的功能。	7.2节
锁存功能	电源断、复位操作时保持软元件数据的功能。	7.3节
停止→运行时的输出状态的选择功能	选择基本型QCPU从停止状态转为运行状态时的输出（Y）的状态（停止前的输出的重新输出/运算执行后的输出）的功能。	7.4节
时钟功能	启动基本型QCPU内置时钟的功能。	7.5节
远程操作	从远处操作基本型QCPU的功能	7.6节
远程运行/停止	反复停止、启动顺控程序运算的功能。	7.6.1节
远程暂停	在保持基本型QCPU的状态下，停止顺控程序运算的功能。	7.6.2节
远程复位	基本型QCPU处于停止状态时复位的功能。	7.6.3节
远程锁存清零	基本型QCPU处于停止状态时锁存数据清零的功能。	7.6.4节
Q系列对应输入模块的输入响应时间选择	从1 ms、5 ms、10 ms、20 ms、70 ms中选择Q系列对应输入模块的输入响应时间的功能。（默认值：10 ms）	7.7.1节
Q系列对应高速输入模块的输入响应时间选择	从0.1 ms、0.2 ms、0.4 ms、0.6 ms、1 ms中选择Q系列对应高速输入模块的输入响应时间的功能。（默认值：0.2 ms）	7.7.2节
Q系列对应中断模块的输入响应时间选择	将Q系列对应中断模块的输入响应时间变更为0.1 ms、0.2 ms、0.4 ms、0.6 ms、1 ms的功能。（默认值：0.2 ms）	7.7.3节
Q系列对应智能功能模块的开关设置	对智能功能模块进行各种设置的功能。 （设置内容可参照各智能功能模块）	7.8节
运行中的写入	基本型QCPU处于运行过程中的程序写入功能。	7.9节
警戒定时器	对基本型QCPU因硬件、程序异常而引起的运算迟缓进行监视的功能。	7.11节
自我诊断功能	基本型QCPU诊断自身有无异常的功能。	7.12节
故障记录	将自我诊断结果作为故障记录存储在存储器内的功能。	7.13节
系统保护	防止来自GX Developer、串行通信模块等的程序更改的功能。	7.14节
口令注册	禁止从GX Developer对基本型QCPU的各文件进行读出/写入的功能。	7.14.1节
系统显示	连接GX Developer，对系统构成进行监视的功能。	7.15节
LED显示	利用基本型QCPU前面的LED显示基本型QCPU动作状态的功能。	7.16节
串行通信功能	Q00CPU/Q01CPU的RS—232接口和个人电脑/显示器等的RS—232接口相连接，采用MC协议进行通信的功能。	7.17节

7.2 恒定扫描

（1）什么是恒定扫描

扫描时间根据顺控程序所使用的指令的执行/非执行其处理时间是不同的，因此，每次扫描并不相同，而是变化的。

恒定扫描就是在保持扫描时间为一定时间的同时，反复执行顺控程序的功能。

此外，由于输入/输出刷新是在顺控程序执行前进行的，因此，即使顺控程序的执行时间发生变化，通过采用恒定扫描功能，可以使输入/输出刷新的间隔保持一定。

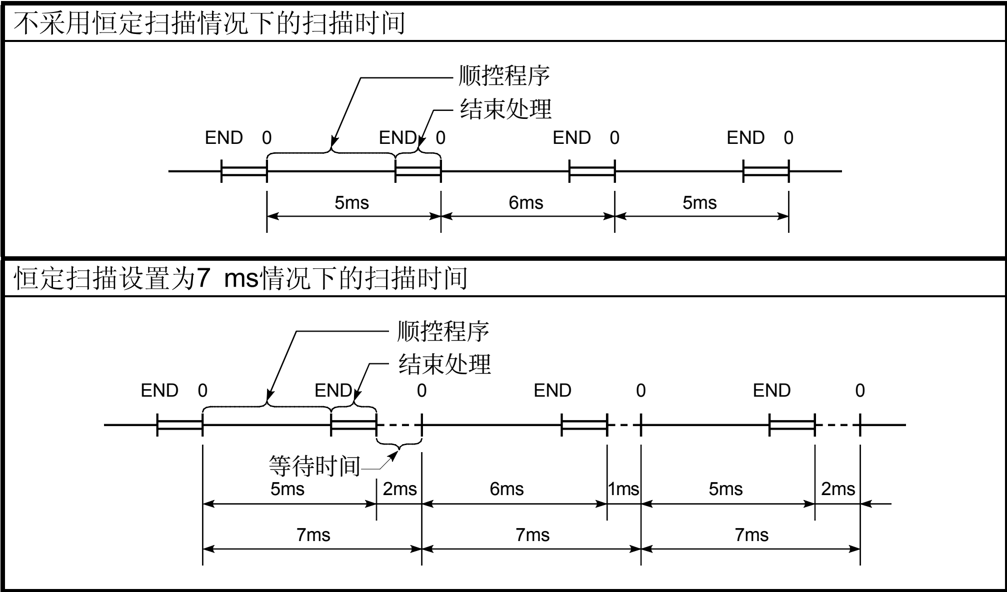
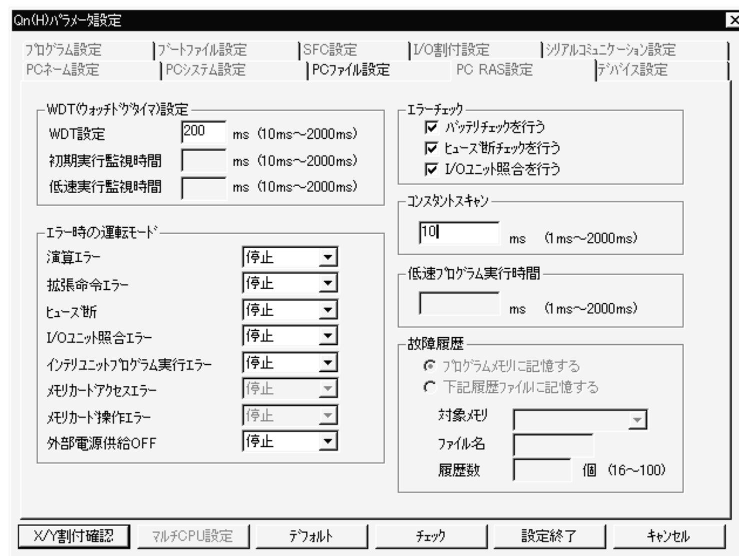


图7.1 恒定扫描的动作

（2）恒定扫描时间的设置

（a）恒定扫描时间的设置采用PC参数的PC RAS设置进行。
恒定扫描时间可以在1～2000 ms的范围内进行设置。
（设置单位为1ms。）
- 执行恒定扫描的情况下，设置恒定扫描时间。
- 不执行恒定扫描的情况下，恒定扫描时间设置为“空”。

（例） 恒定扫描时间设置为10 ms的场合



（b）恒定扫描的设置时间请在长于顺控程序的最大扫描时间并短于WDT设置时间的范围内进行设置。

（WDT设置时间）>（恒定扫描设置时间）>（顺控程序的最大扫描时间）

顺控程序的扫描时间长于恒定扫描设置时间的情况下，基本型将检测出PRG. TIME OVER（出错码：5010），忽略恒定扫描时间而执行顺控程序的扫描。

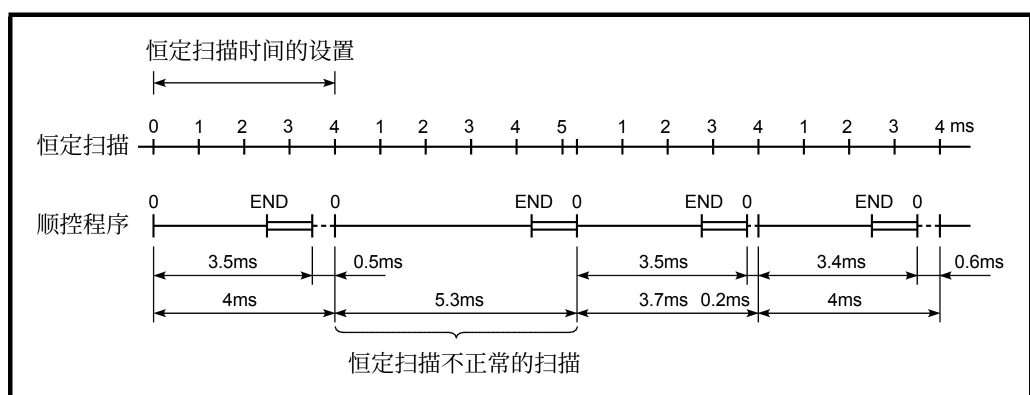


图7.2 扫描时间长于恒定扫描情况下的动作

如果长于WDT设置时间，基本型QCPU就会检测出WDT出错，停止程序的执行。

（c）从顺控程序执行结束处理到下一次扫描开始为止的等待期间，将中断顺控程序的处理。
但是，如果结束处理后发生中断原因，则执行中断程序。

（d）恒定扫描的精度
此处阐述恒定扫描时间设置时的精度的有关内容。
① 设置恒定扫描时间，未执行中断程序时的误差的有关内容请参照第11章。
② 恒定扫描的等待期间也执行中断程序。
由于中断程序执行期间未中断禁止，因此，中断程序执行过程中即使到达了恒定扫描时间，在中断程序结束之前，也不能结束恒定扫描。
使用中断程序的情况下，仅在中断程序执行期间恒定扫描时间有可能会发生偏差。

备　注

所使用的指令的处理时间的有关内容请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

7.3 锁存功能

（1）什么是锁存功能

（a）基本型QCPU的各个软元件在以下情况下将被清零，成为默认值（位软元件：断、字软元件：0）。

- 可编程控制器的电源切断时
- 复位操作时
- 允许时间以上的瞬间掉电时

锁存功能就是在上述情况下仍能保持软元件内容的功能。

程序中的运算不管有/无锁存，都是相同的。

（b）锁存可用于连续控制的生产个数、次品数、地址等数据的管理过程中，即使出现可编程控制器的电源切断、复位操作、允许时间以上的瞬间掉电等情况，仍须保持上述数据并继续控制的场合。

（c）可以锁存的是如下所示的软元件。

（默认的锁存范围设置只有锁存继电器。）

① 锁存继电器（L）

② 通信继电器（B）

③ 信号报警器（F）

④ 边沿继电器（V）

⑤ 定时器（T）

⑥ 累加定时器（ST）

⑦ 计数器（C）

⑧ 数据寄存器（D）

⑨ 通信寄存器（W）

（2）锁存范围的设置

锁存范围的设置在PC参数的软元件设置时进行

锁存范围设置有锁存清零（远处锁存清零操作）有效的范围和无效的范围2种。

Qn(H)ﾊﾟﾗﾒｰﾀ設定

ﾌﾟﾛｸﾞﾗﾑ設定 | ﾌﾞｰﾄﾌｧｲﾙ設定 | SFC設定 | I/O割付設定 | ｼﾘｱﾙｺﾐｭﾆｹｰｼｮﾝ設定

PCﾈｰﾑ設定 | PCｼｽﾃﾑ設定 | PCﾌｧｲﾙ設定 | PC RAS設定 | ﾃﾞﾊﾞｲｽ設定

	記号	進	ﾃﾞﾊﾞｲｽ点数	ﾗｯﾁ(1)先頭	ﾗｯﾁ(1)最終	ﾗｯﾁ(2)先頭	ﾗｯﾁ(2)最終	ﾛｰｶﾙﾃﾞﾊﾞｲｽ先頭	ﾛｰｶﾙﾃﾞﾊﾞｲｽ最終
入力ﾘﾚｰ	X	16	2K						
出力ﾘﾚｰ	Y	16	2K						
内部ﾘﾚｰ	M	10	8K						
ﾗｯﾁﾘﾚｰ	L	10	2K						
ﾘﾝｸﾘﾚｰ	B	16	2K						
ｱﾅﾝｼｪｰﾀ	F	10	1K						
ﾘﾝｸ特殊	SB	16	1K						
ｴｯｼﾞﾘﾚｰ	V	10	1K						
ｽﾃｯﾌﾟﾘﾚｰ	S	10	2K						
ﾀｲﾏ	T	10	512						
積算ﾀｲﾏ	ST	10	0K						
ｶｳﾝﾀ	C	10	512						
ﾃﾞｰﾀﾚｼﾞｽﾀ	D	10	11136						
ﾘﾝｸﾚｼﾞｽﾀ	W	16	2K						
ﾘﾝｸ特殊	SW	16	1K						

ﾃﾞﾊﾞｲｽ合計 16.4 Kﾜｰﾄﾞ

ﾜｰﾄﾞﾃﾞﾊﾞｲｽ 15.2 Kﾜｰﾄﾞ

ﾋﾞｯﾄﾃﾞﾊﾞｲｽ 19.0 Kﾋﾞｯﾄ

ﾃﾞﾊﾞｲｽ点数の合計は16704ﾜｰﾄﾞまでです。
ﾗｯﾁ(1):ﾗｯﾁｸﾘｱにてｸﾘｱが可能です。
ﾗｯﾁ(2):ﾗｯﾁｸﾘｱにてｸﾘｱが不可です。
PCﾒﾓﾘｸﾘｱにてｸﾘｱを行ってください。

高速ﾃﾞﾊﾞｲｽｴﾘｱ X0～C511

X/Y割付確認 | ﾏﾙﾁCPU設定 | ﾃﾞﾌｫﾙﾄ | ﾁｪｯｸ | 設定終了 | ｷｬﾝｾﾙ

（3）锁存范围的软元件数据的清零

进行锁存清零情况下的软元件状态如下表所示。

锁存的有无	采用锁存清零的清零/保持
未进行锁存范围指定的软元件	清零
锁存（1）设置（设置为“可以用锁存清零进行清零”的软元件）	清零
锁存（2）设置（设置为“不可用锁存清零进行清零”的软元件）	保持

要　　点

文件寄存器（R）即使进行锁存清零也无法清零。
文件寄存器清零的有关内容请参照10.7节。

（4）注意事项

锁存范围的软元件内容靠基本型QCPU上安装的电池（Q6BAT）来保持。

（a）即使在顺控程序用标准ROM实现ROM化并进行ROM运行的情况下，如果需要锁存，仍须用到电池。

（b）顺控程序的电源处于切断状态时，如果将电池的接插件从基本型QCPU的接插件上拔下，锁存范围的软元件内容将不再保持而变为不定值，请加以注意。

7.4 停止状态←→运行状态时的输出（Y）状态的设置

（1）从停止状态转为运行状态时的输出（Y）状态的设置

如果从运行状态转为停止状态，运行状态的输出（Y）就存储在可编程控制器内部，输出（Y）全部切断。

在基本型QCPU上，从停止状态转为运行状态时，可以从以下2种之中选择状态。

- 输出停止前的输出状态
- 输出清零

（默认为从停止状态转为运行状态时，输出停止前的输出（Y）状态后，执行程序。）

（a）输出停止前的输出（Y）状态

将即将进入停止状态前的输出（Y）状态输出后，进行顺控程序的运算。

（b）输出清零（1次扫描后输出）

清除所有的输出（Y）并执行顺控程序的运算后，进行输出（Y）的输出。

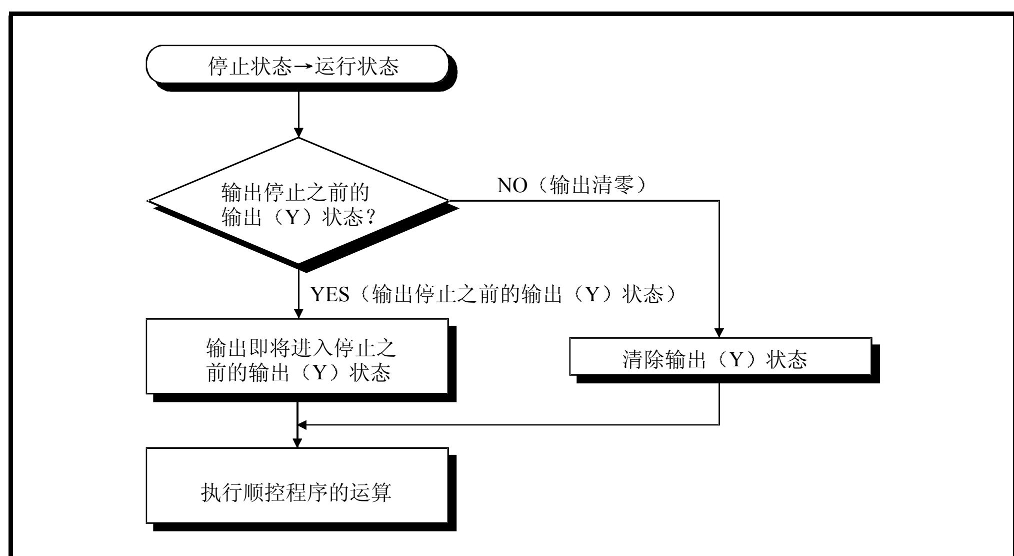


图7.3 从停止状态→运行状态时的处理

（2）从运行状态转为停止状态时的输出（Y）状态的设置

从停止状态转为运行状态时，停止前的输出状态可以由PC参数的PC系统设置进行设置。

停止→运行时的输出模式的设置

（3）注意事项

基本型QCPU处于停止状态时即使强制性地接通输出（Y），一旦从停止状态转换到运行状态，就不能保持接通的状态。

此时，就进入PC系统设置的停止→运行时的输出模式下所设置的输出状态。

7.5 时钟功能

（1）什么是时钟功能

（a）时钟功能就是用顺控程序读出基本型QCPU内部的时钟数据并用于时间管理的功能。

此外，时钟数据也可以用于故障记录中的日期存储等由系统完成的功能的时间管理。

时钟的运行即使在可编程控制器的电源切断或发生允许时间以上的瞬间掉电时仍可靠基本型QCPU内部的电池（Q6BAT）来继续。

（b）时钟数据

所谓时钟数据，就是基本型QCPU内部的时钟元件所使用的数据，如下表所示。

数据名称	内　　容	
年	公历4位（可在1980年～2079年的范围内计测）	
月	1～12	
日	1～31（闰年自动判别）	
时	0～23（24小时制）	
分	0～59	
秒	0～59	
星期	0	星期日
	1	星期一
	2	星期二
	3	星期三
	4	星期四
	5	星期五
	6	星期六

（2）时钟数据对时钟元件写入和读出

（a）时钟数据对时钟元件的写入有2种方法，即“用GX Developer进行的方法”和“用程序进行的方法”。

① 用GX Developer进行的方法

使用GX Developer的情况下，用“在线”→“时钟设置” 操作显示时钟设置窗口，将时钟数据写入时钟元件。

時計設定

接続先情報
接続ｲﾝﾀｰﾌｪｰｽ COM1 ←→ CPUﾕﾆｯﾄ
接続先PC ﾈｯﾄﾜｰｸNO. 0 局番 自局 PCﾀｲﾌﾟ Q01

時計設定
年 月 日 時 分 秒 曜日
2001 05 29 08 38 53 火曜日

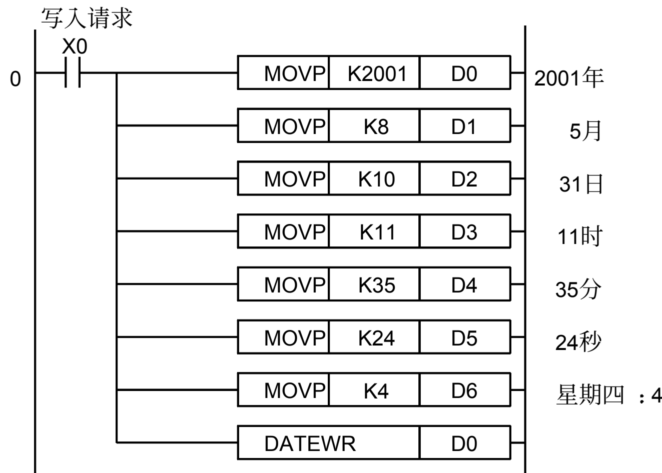
実行先指定
現在指定局 ｸﾞﾙｰﾌﾟNo. 1
実行ﾕﾆｯﾄ指定
1 枚目

設定　閉じる

② 用程序进行的方法

在程序中通过时钟数据的写入指令（DATEWR）将时钟数据写入基本型QCPU的时钟元件中。

时钟数据的写入指令（DATEWR）中设置为D0～D6的时钟数据写入程序如下图所示。



DATEWR的细节请参照下列手册。

• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

（b）时钟数据的读出

需要将时钟数据读入数据寄存器的情况下，请在程序中使用时钟数据的读出指令（DATERD）。

将时钟数据读出指令所读出的时钟数据存储到D0～D6的程序如下图所示。



DATERD指令的细节请参照QCPU（Q模块）/QnACPU编程手册（通用指令篇）。

备　注

1） 时钟数据的写入/读出也可以利用特殊继电器（SM210～SM213）和特殊寄存器（SD210～SD213）进行。

特殊继电器的详情请参照附录1；特殊寄存器的详情请参照附录2。

2） *：D10～D16中存储的数据如下图所示。

软元件	数据	内容	
D10	2001	公历 4 位	参照 7.5节之(1)
D11	5	月	
D12	31	日	
D13	11	时	
D14	35	分	
D15	24	秒	
D16	4	星期	

（3）注意事项

（a）时钟数据出厂时未设置。

时钟数据是基本型QCPU的系统以及智能功能模块用于故障记录等用途的，因此，初次使用基本型QCPU时，请务必设置时间。

（b）即使是只修正部分时钟数据的情况下，也必须将所有数据重新写入时钟元件。

（c）写入时钟元件中的数据在7.5节(1)的(b)中所示的范围内进行数据检查。

因此，如果写入了7.5节(1)的(b)中所示的范围内不可能作为时刻的数据，就不能进行正常的时钟动作。

例

	写入时钟因素的动作	CPU模块的动作状态
2月30日	执行	DATEW 指 令 执 行 时 ： OPERATION ERROR（操作出错） （出错代码：4100） SM210 ON时：SM211为通
13月32日	不执行	不能检测出出错

（4）时钟数据的精度

时钟功能的精度根据环境温度有所不同，具体如下：

环境温度（℃）	精度（日误差，S）
0	－3.2 ～ ＋5.27（TYP.＋1.98）
＋25	－2.57 ～ ＋5.27（TYP.＋2.22）
＋55	－11.68 ～ ＋3.65（TYP.－2.64）

（5）时钟数据的比较

读出时钟数据并用顺控程序进行比较时，请使用时钟数据读出指令（DATERD）读出时钟数据。

年的数据采用公历4位读出，因此，可以原封不动地采用比较指令进行比较。

7.6 远程操作

基本型QCPU可以通过运行/停止/复位开关进行停止状态和运行状态的转换。
此外，还可以利用运行/停止/复位开关进行复位。

远程操作就是通过外部（GX Developer、智能功能模块、远程触点等）的操作对基本型QCPU的动作进行控制。

远程操作有以下4种：
- 远程运行/停止
- 远程暂停
- 远程复位
- 远程锁存清零

备　注

本项的智能功能模块的说明以串行通信模块为例。

7.6.1 远程运行/停止

（1）什么是远程运行/停止

（a）远程运行/停止就是将运行/停止/复位开关保持在运行的位置上，并从外部运行/停止基本型QCPU。

（b）在如下的情况下，利用远程操作进行远程运行/停止将是非常方便的。

① 基本型QCPU处于手无法触及的位置时。

② 控制箱内的基本型QCPU通过外部信号运行/停止时。

（c）远程运行/停止时的运算
进行远程运行/停止的程序运算如下。

① 远程停止
程序执行到END指令为止，进入停止状态。

② 远程运行
如果远程停止处于停止状态时进行远程运行，则重新进入运行状态，程序从第0步开始执行。

（2）远程运行/停止的方法

远程运行/停止的方法有2种，即“通过远程运行触点的方法”和“通过GX Developer、串行通信模块等的方法”。

（a）通过远程运行触点的方法

远程运行触点在PC参数的PC系统设置时进行设置。

可以设置的软元件范围为：输入X0～7FF。

通过设置远程运行触点的通/断，就能进行远程运行/停止。
① 远程运行触点为断的情况下，CPU进入运行状态。
② 远程运行触点为通的情况下，CPU进入停止状态。

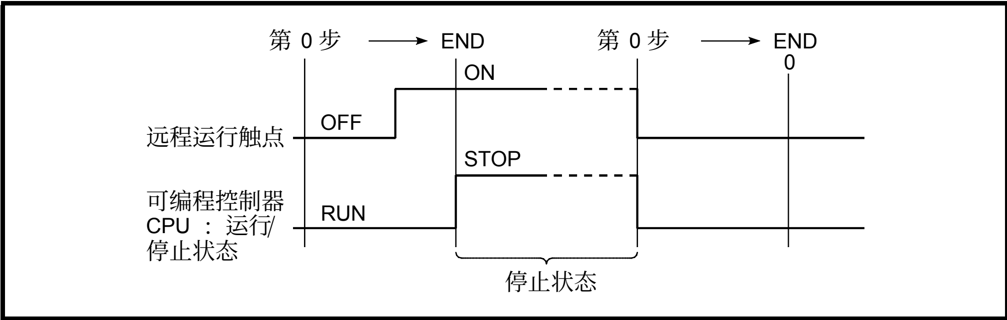


图7.4 通过远程运行触点的运行/停止

（b）通过GX Developer、串行通信模块等的方法

利用来自GX Developer、串行通信模块、Ethernet接口模块等的远程运行/停止操作，可以实现基本型QCPU的运行/停止。
GX Developer的操作采用在线远程操作进行。

串行通信模块、Ethernet接口模块上的控制采用MC协议的指令进行。
MC协议的详细内容请参照下列手册。
• Q对应MELSEC通信协议参考手册。

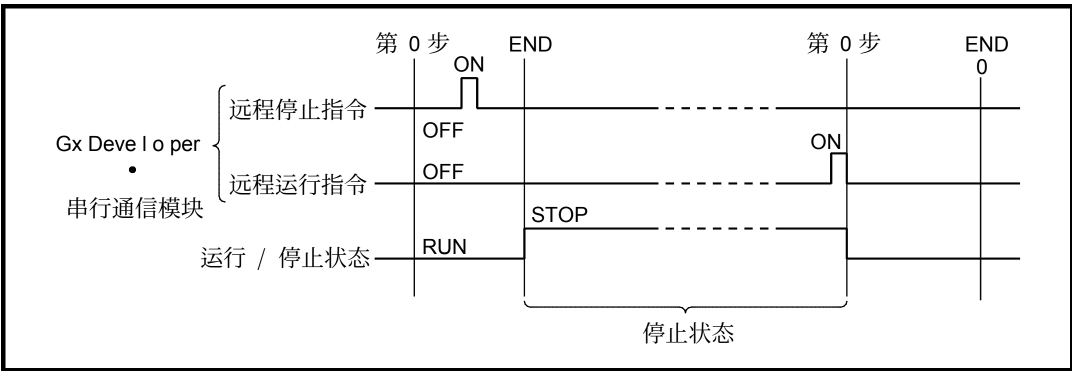


图7.5 通过GX Developer、串行通信模块等的远程运行/停止

（3）注意事项

由于基本型QCPU为停止优先，因此，请注意以下几点。

（a）基本型QCPU通过远程运行触点、GX Developer、串行通信模块等之中的任何一种进行远程停止都将进入停止状态。

（b）由最初进行远程停止的外部原因（远程运行触点、GX Developer、串行通信模块等）进行远程运行，就进入运行状态。

备　注

运行/停止状态如下所示。

- 运行状态……………………反复执行顺控程序的第0步～END/FEND指令的运算的状态。
- 停止状态……………………顺控程序的运算处于停止中的状态，输出（Y）全部切断。

7.6.2 远程暂停

（1）什么是远程暂停

（a）远程暂停是将运行/停止/复位开关保持在运行的位置上，并从外部使基本型QCPU进入暂停状态。
所谓暂停状态，就是在保持所有输出（Y）的通/断状态原封不动的情况下，停止基本型QCPU的运算。

（b）过程控制等中，有时希望将基本型QCPU在运行状态下设置为通的输出（Y）在变为停止状态后仍保持通不变，远程暂停恰好适合这样的场合。

要　　点
发生停止出错时，输出（Y）为断。 发生停止出错时仍希望保持输出的情况下，请在PC参数的输入/输出分配时进行输出的保持设置。

（2）远程暂停的方法
远程暂停的方法有2种，即“通过远程暂停触点的方法”和“通过GX Developer、串行通信模块等的方法”。

（a）通过远程暂停触点的方法
远程暂停触点在PC参数的PC系统设置时进行设置。
不可仅设置远程暂停触点。
设置远程暂停触点时，远程运行触点也请一并设置。

可以设置的软元件范围为：输入X0～7FF。

① 同时接通远程暂停和暂停允许插头（SM206）的扫描的结束处理执行时，暂停状态触点（SM204）接通。
如果暂停状态触点接通以后的下一次扫描一直执行到END指令为止，就进入暂停状态，停止运算。

② 切断远程暂停触点，或切断SM206，暂停状态就被解除，重新从第0步开始进行顺控程序的运算。

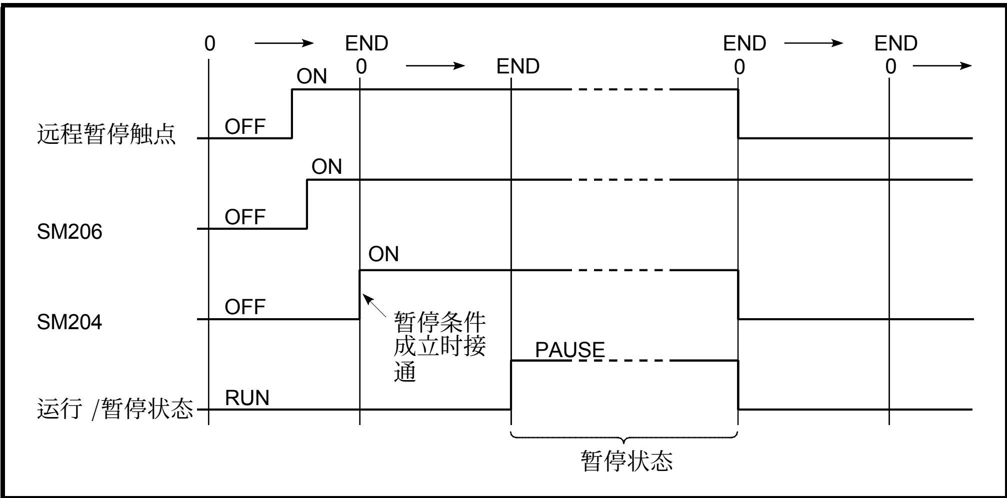


图7.6　通过远程暂停触点暂停时的时序图

（b）通过GX Developer、串行通信模块等的方法
可以从GX Developer、串行通信模块等进行远程暂停操作。
GX Developer的操作采用在线远程操作进行。

串行通信模块、Ethernet接口模块上的控制采用MC协议的指令进行。
MC协议的详细内容请参照下列手册。
• Q对应MELSEC通信协议参考手册。

① 包含远程暂停指令的扫描的结束处理执行时，接同暂停状态触点（SM204）。
② 如果暂停状态触点接通以后的下一次扫描一直执行到END指令为止，就进入暂停状态，停止运算。

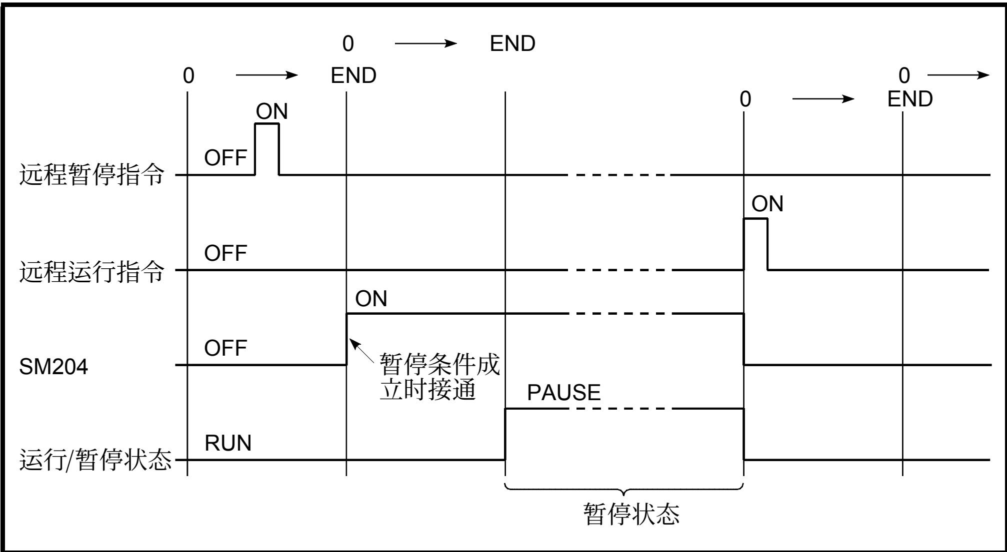
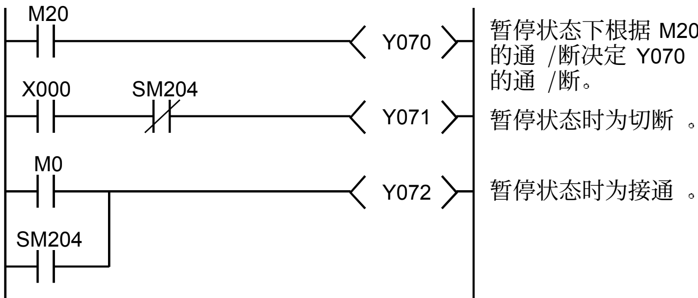


图7.7　通过GX Developer暂停时的时序图

（3）注意事项
设为暂停状态时，需要预置输出（Y）通/断状态的，请使用暂停状态触点（SM204）进行连锁。



7.6.3 远程复位（REMOTE RESET）

（1）什么是远程复位

（a）远程复位就是基本型QCPU处于停止状态时，通过来自外部的操作对基本型QCPU进行复位。

此外，即使运行/停止/复位开关处于“运行”位置，因发生自我诊断可以检测出的出错而基本型QCPU处于停止状态时，可以复位。

（b）远程复位可以在无法直接操作基本型QCPU的运行/停止/复位开关的部位发生出错时，利用远程操作对基本型QCPU进行复位。

由于远程复位只可在停止状态时执行，因此，基本型QCPU处于运行状态时，须利用远程停止设为停止状态。

（2）远程复位的方法

远程复位只能从GX Developer、串行通信模块上进行操作。

远程复位按以下步骤进行。

（a）PC参数的系统设置时将远程复位设置为“允许”，并向基本型QCPU写入PC参数。



（b）基本型QCPU处于运行状态时，利用远程停止操作设置为停止状态。

（c）利用远程复位操作对基本型QCPU进行复位。

① 采用GX Developer时，进行在线远程操作。

② 采用串行通信模块、Ethernet接口模块时，利用MC协议的指令进行。

MC协议的详细内容请参照下列手册。

- Q对应MELSEC通信协议参考手册

（3）注意事项

（a）进行远程复位时，请在PC参数的系统设置时将远程复位设置为“允许”，并将PC参数写入基本型QCPU。
远程复位未设置为“允许”的情况下，不能进行远程复位。

（b）基本型QCPU处于运行状态时，不能通过远程复位进行复位。
请先利用远程停止操作等将基本型QCPU设为停止状态后，再进行远程复位操作。

（c）利用远程复位的复位处理完成后，基本型QCPU就进入运行/停止/复位开关所设置的运行状态。
① 运行/停止/复位开关处于“停止”位置时，复位完成后基本型QCPU就进入停止状态。
② 运行/停止/复位开关处于“运行/”位置时，复位完成后基本型QCPU就进入运行/状态。

（d）基本型QCPU因噪音等发生异常的情况下，有时可能无法通过远程复位进行复位，请加以注意。
无法用远程复位复位时，或者用运行/停止/复位开关复位，或者重新开启可编程控制器的电源。

要　点

（1）如果基本型QCPU因出错而停止时进行远程复位，则在复位结束时基本型QCPU会进入由运行/停止/复位开关设定的运行状态，请加注意。
（2）PC参数的系统设置时即使未设置远程复位为“允许”，GX Developer的远程复位处理也会完成。
但是，由于基本型QCPU上未进行复位处理，因此，将不会复位。
利用GX Developer进行远程复位后基本型QCPU的状态仍不改变的情况下，请在PC参数的系统设置时确认远程复位是否为“允许”。

7.6.4 远程锁存清零

（1）什么是远程锁存清零
远程锁存清零就是基本型QCPU处于停止状态时，通过GX Developer等将锁存的软元件数据复位。

（2）远程锁存清零的方法
远程锁存清零只能从GX Developer、串行通信模块等进行操作。
远程锁存清零按以下步骤进行。

（a）利用运行/停止/复位开关或远程停止将基本型QCPU设为停止状态。

（b）利用远程锁存清零操作对基本型QCPU进行锁存清零

① GX Developer的操作采用在线远程操作进行。

② 串行通信模块、Ethernet接口模块上的控制采用MC协议的指令进行。
MC协议的详细内容请参照下列手册。
• Q对应MELSEC通信协议参考手册

（c）远程锁存清零操作结束后，如果要进入运行状态，请将运行/停止/复位开关设到“运行”位置上，或进行远程运行操作。

（3）注意事项

（a）基本型QCPU处于运行状态时，不可通过远程锁存清零操作进行锁存清零。

（b）参数模式的软元件设置时所设置的软元件的锁存范围内，包括锁存清零（远程锁存清零操作）有效的范围和无效的范围。
远程锁存清零操作时，只有设置为“锁存清零有效”范围内的软元件范围被复位。
关于设置为锁存清零无效的软元件的复位方法，请参照 4.6节。

（c）远程锁存清零执行时，未锁存的软元件也将被清零。
基本型QCPU的故障记录保存存储器的数据在远程锁存清零时不会被清零。

7.6.5 远程操作和基本型QCPU的运行/停止状态之间的关系

（1）远程操作和基本型QCPU的运行/停止状态之间的关系
通过远程操作和基本型QCPU的运行/停止状态组合的基本型QCPU的动作如下表所示。

远程操作 \ 运行/停止/复位开关	运行＊1	停止	暂停＊2	复位＊3	锁存清零
运行	运行	停止	暂停	不可操作＊4	不可操作＊4
停止	停止	停止	停止	复位＊5	锁存清零

＊1 采用远程运行触点进行的情况下，必须在PC参数的PC系统设置时设置“运行→暂停触点”。
＊2 采用远程暂停触点进行的情况下，必须在PC参数的PC系统设置时设置“运行→暂停触点”。而且，必须事先接通远程暂停允许线圈（SM206）。
＊3 必须在PC参数的PC系统设置时将“远程复位”设置为“允许”。
＊4 基本型QCPU通过远程停止操作设为停止状态的情况下，可以进行复位或锁存清零。
＊5 也包括基本型QCPU因故障而处于停止状态的情况。

（2）来自同一GX Developer的远程操作
利用同一个GX Developer进行远程操作的情况下，进入后面执行的远程操作的状态。

（3）来自多个GX Developer的远程操作
利用一个GX Developer进行远程操作的基本型QCPU不可再从其他GX Developer上进行远程操作。
从正在进行远程操作的GX Developer上进行远程运行，再解除远程操作后，其他GX Developer方可进行远程操作。
例如，正在从某台GX Developer上进行远程暂停的情况下，即使从其他GX Developer上进行远程停止/远程运行，仍将保持暂停状态不变。
从正在进行远程暂停的GX Developer进行远程运行，再解除远程操作后，方可从其他GX Developer上进行远程操作。

7.7 Q系列对应输入模块的输入响应速度的选择（输入/输出响应时间）

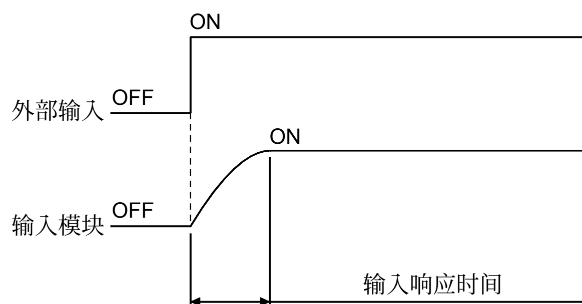
7.7.1 输入模块的输入响应时间的选择

（1）什么是输入模块的输入响应时间选择

Q系列对应输入模块的输入响应速度可以以模块为单位在1 ms，5 ms，10 ms，20 ms，70 ms之间变化。

输入模块采用所设置的输入响应速度接收外部输入。

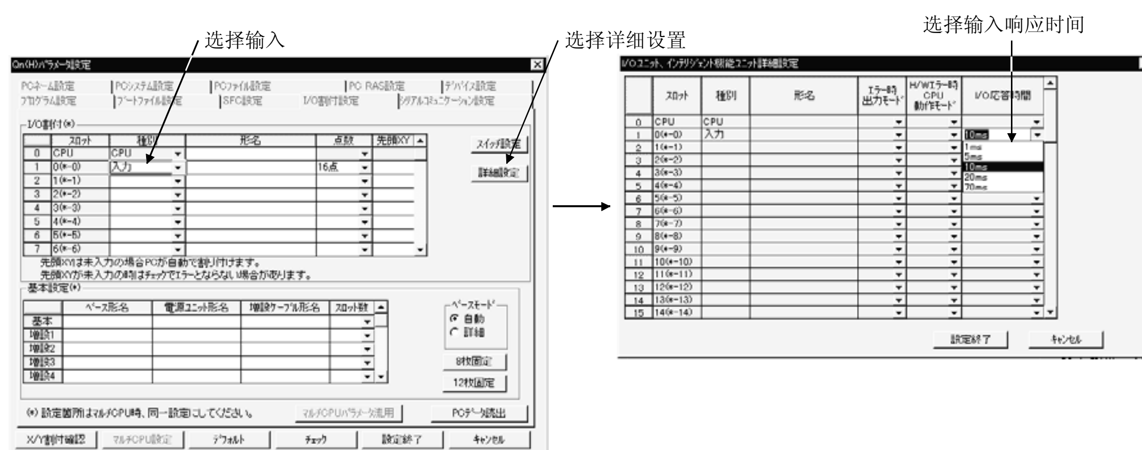
输入响应速度的默认值设置为10 ms。



（2）输入响应速度的选择

输入响应速度的设置在PC参数的输入/输出分配时进行。

设置输入响应速度的插槽的类别选择“输入”。



（3）注意事项

（a）输入响应速度设置为高速时，容易受到噪音等的干扰。

输入响应速度请考虑使用环境后再设置。

（b）输入响应速度的设置在以下情况下有效。

- 可编程控制器的电源接通时
- 基本型QCPU复位时

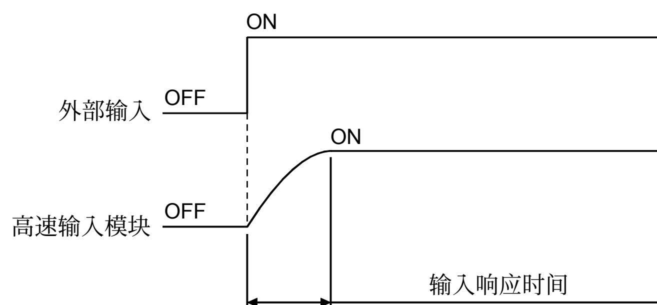
7.7.2 高速输入模块的输入响应速度的选择

（1）什么是高速输入模块的输入响应速度选择

Q系列对应高速输入模块（QX40—S1）的输入响应速度可以以模块为单位在0.1 ms，0.2 ms，0.4 ms，0.6 ms，1 ms之间变更。

高速输入模块采用所设置的输入响应速度接收外部输入。

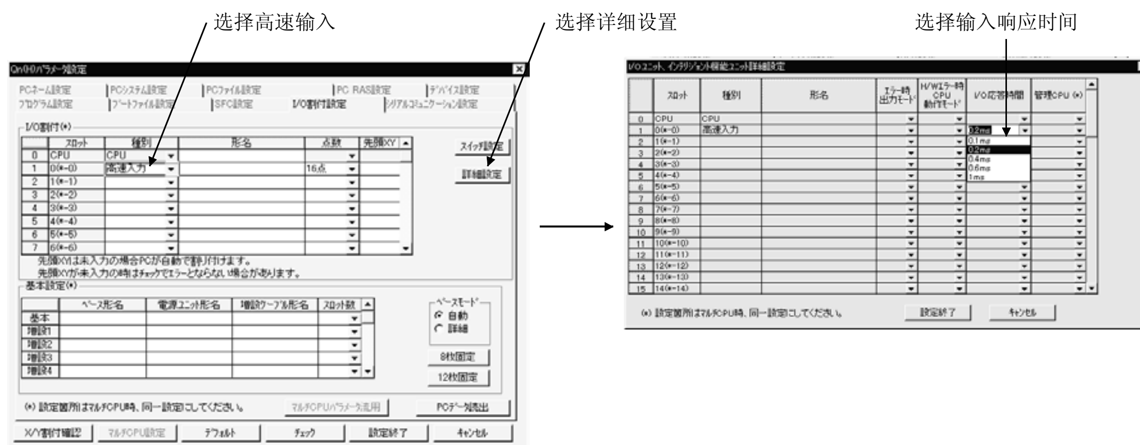
输入响应速度的默认值设置为0.2 ms。



（2）输入响应速度的设置

输入响应速度的设置在PC参数的输入/输出分配时进行。

设置输入响应速度的插槽的类别选择“高速输入”。



（3）注意事项

（a）输入响应速度设置为高速时，容易受到噪音等的干扰。

输入响应速度请考虑使用环境后再设置。

（b）输入响应速度的设置在以下情况下有效。

• 可编程控制器的电源接通时
• 基本型QCPU复位时

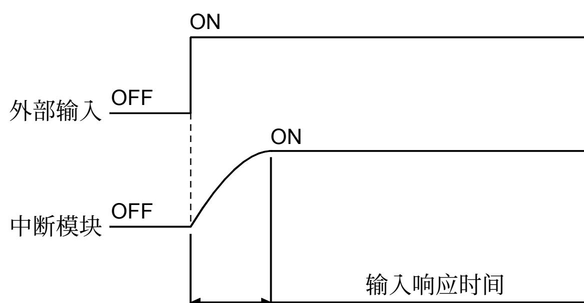
7.7.3 中断输入模块的输入响应速度的选择

（1）什么是中断输入模块的输入响应速度选择

Q系列对应的中断模块（QI16）的输入响应速度可以以模块为单位在0.1 ms，0.2 ms，0.4 ms，0.6 ms，1 ms之间变化。

中断模块采用所设置的输入响应速度接收外部输入。

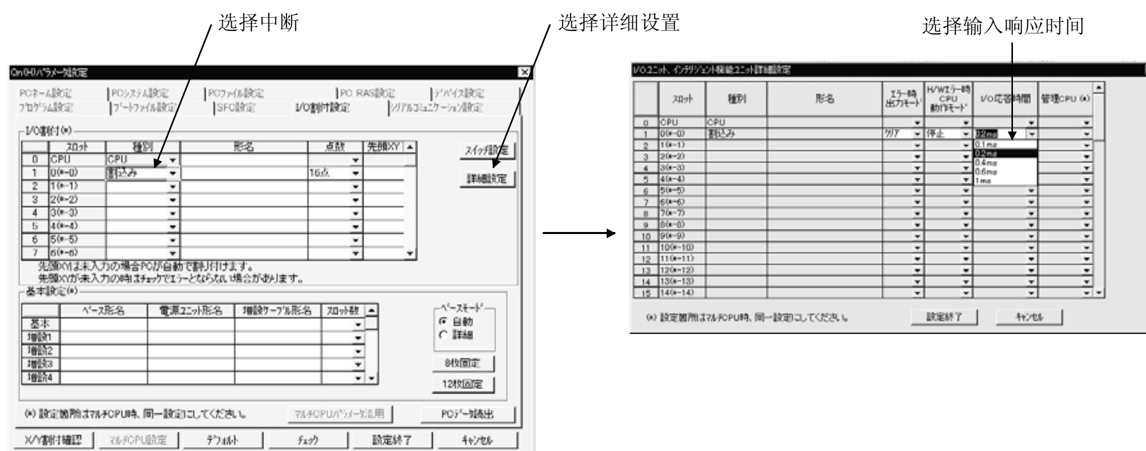
输入响应速度的默认值设置为0.2 ms。



（2）输入响应速度的设置

输入响应速度的设置在PC参数的输入/输出分配时进行。

设置输入响应速度的插槽的类别选择“中断”。



（3）注意事项

（a）输入响应速度设置为高速时，容易受到噪音等的干扰。

输入响应速度请考虑使用环境后再设置。

（b）输入响应速度的设置在以下情况下有效。

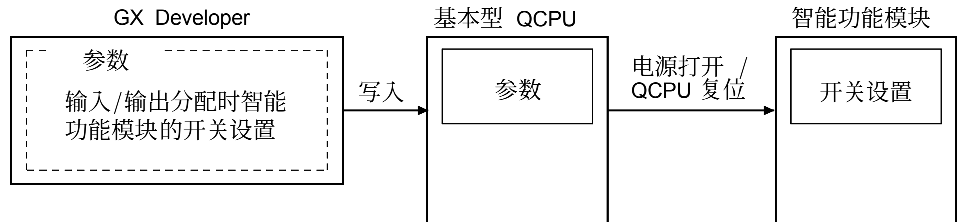
- 可编程控制器的电源接通时
- 基本型QCPU复位时

7.8 智能功能模块的开关设置

（1）什么是智能功能模块的开关设置

智能功能模块的开关设置就是利用GX Developer设置Q系列对应的智能功能模块的开关内容。

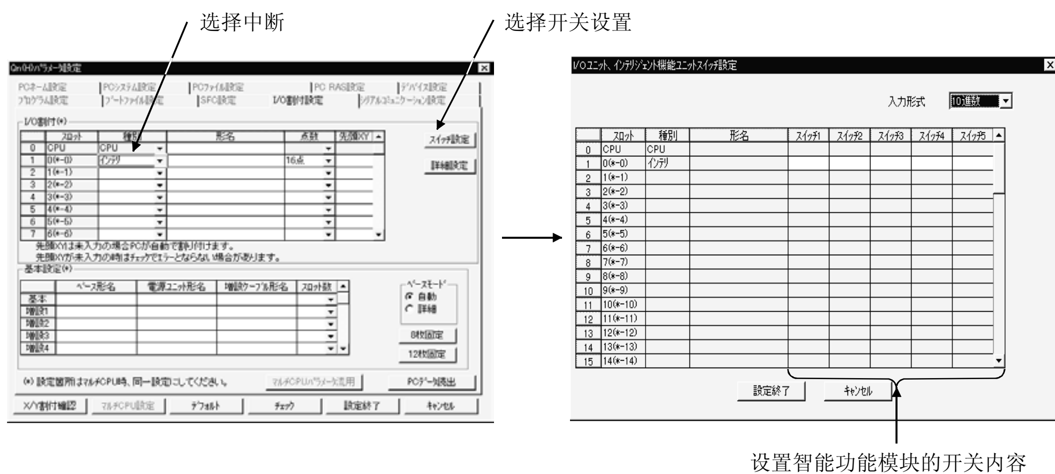
所设置的开关设置在可编程控制器的电源打开时或基本型QCPU复位时从基本型QCPU写入各智能功能模块。



（2）智能功能模块的开关设置

智能功能模块的开关设置在PC参数的输入/输出分配时进行。

智能功能模块的开关设置在PC参数的输入/输出分配时进行。



（3）注意事项

（a）智能功能模块的开关设置内容请参照所使用的智能功能模块的手册。

（b）采用GX Developer中断模块的开关设置须将类别设置为“中断”后方可进行。
关于中断模块的开关设置的细节，请参见下列手册。
- 构件型输入输出模块用户手册。

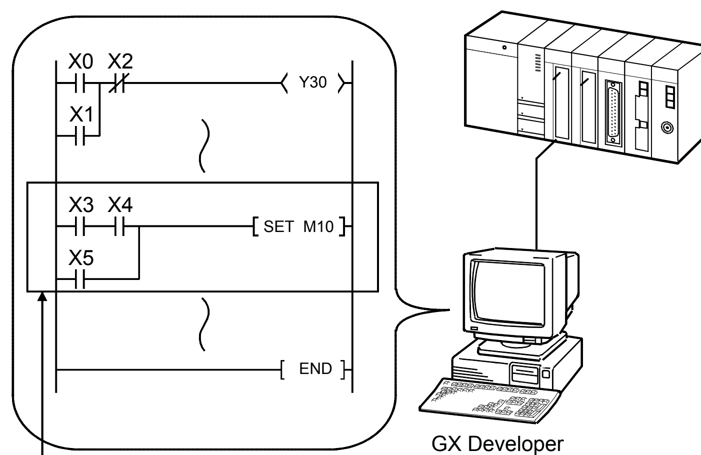
（c）开关设置在以下情况下有效。
- 可编程控制器的电源接通时
- 基本型QCPU复位时

7.9 电路模式下的运行中写入

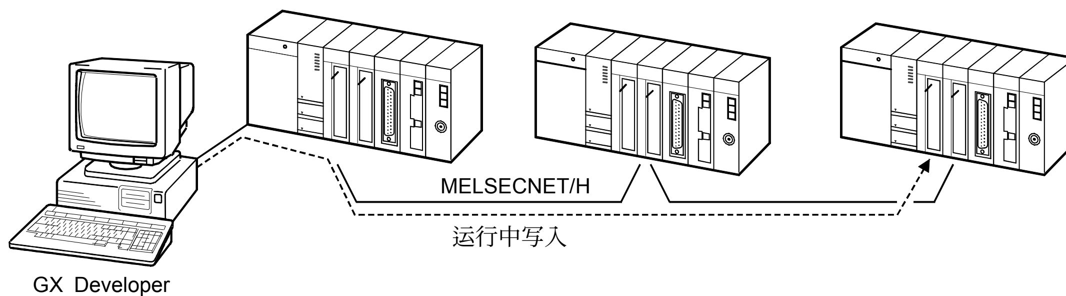
（1）什么是电路模式下的运行中写入

（a）电路模式下的运行中写入就是基本型QCPU处于运行状态时写入程序的功能。

（b）使用电路模式下的运行中写入功能时，可以不用停止基本型QCPU的程序运算就实现程序的更改。



（c）不仅是连接GX Developer的基本型QCPU，网络上连接的其他站的CPU模块的程序也可以进行运行中的写入。



（2）注意事项

运行中写入时的注意事项如下所示。

（a）可以进行运行中写入的只有程序存储器。

从标准ROM的引导运行过程中执行运行中写入的情况下，标准ROM中的程序不会变更。

可编程控制器的电源切断或基本型QCPU复位前，请将程序存储器的内容写入标准ROM。

（b）一次运行中可以写入的步数最大不超过512步。

（c）运行中写入时写入了以下指令的情况下，将不能正常动作。

① 下降沿指令

写入完成时，如果以下的下降沿执行指令的执行条件为断，就执行下降沿执行指令。

- LDF
- ANDF
- ORF
- MEF
- PLF

② 上升沿指令

写入完成时，如果上升沿执行指令（PLS指令/□P指令）的执行条件为通，则不执行上升沿执行指令。

执行条件再次从断→通时，执行上升沿指令。

③ SCJ指令

写入完成时，如果SCJ指令的执行条件为通，则不等待1次扫描，而跳转到指定指针。

7.10 多人监视功能

（1） 什么是多人同时监视功能

（a）可以同时从连接到基本型QCPU、串行通信模块等的多个GX Developer上执行监视。

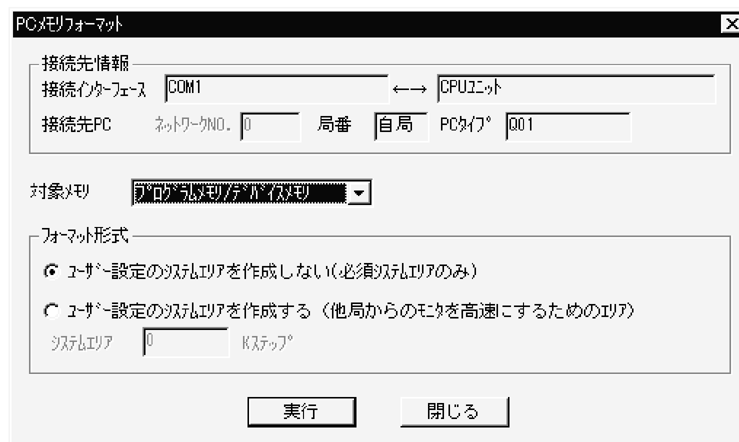
（b）通过建立用户设置的系统区域，可以从多个GX Developer上进行高速监视。（不需要设置本地站的监视文件）

（2） 多人同时高速监视时的设置

（a）多人同时进行高速监视的情况下，按以下步骤建立由用户设置的系统文件。

① 利用在线的PC存储器格式化，显示PC存储器格式化窗口。
② 在对象存储器处选择“程序存储器”。
③ 在格式化形式处设置为“建立由用户设置的系统文件”。
④ 设置系统区域的步数（1k步单位）。

（b）下图所示为系统区域的步数设置为1k步时的设置实例。



① 作为系统区域，可以以1k步为单位，最大设置到3k步。
来自其他站的每1个监视使用1k步。
来自其他站的监视用系统区域最多可设置3个。

（2） 注意事项

（a）即使不事先建立了用户设置的系统区域，也可从其他站同时进行监视，但监视速度将下降。
系统区域设置在程序存储器中，因此，程序的存储区域将减去设置为系统区域的部分。

（b）建立3k用户设置的系统区域的情况下，可以同时从4处对1个CPU进行存取。

7.11 警戒定时器（WDT）

（1） 什么是警戒定时器（WDT）

（a）警戒定时器就是基本型QCPU内部的定时器，用于检测基本型QCPU的硬件和顺控程序的异常。

（b）警戒定时器到达时间时，警戒定时器即出错，基本型QCPU的状态变化如下。

① 全部切断基本型QCPU的输出。

② 正面的运行 LED熄灭，ERR. LED闪烁。

③ SM1接通，SD0中存入出错代码。

（c）警戒定时器的默认值设置为200 ms。
警戒定时器可以在10～2000ms（10ms单位）的范围内变更。

（2） 警戒定时器的设置和复位

（a）警戒定时器的设置时间可以在PC参数的PC RAS设置时更改。

（b）基本型QCPU在结束处理执行过程中将警戒定时器复位。

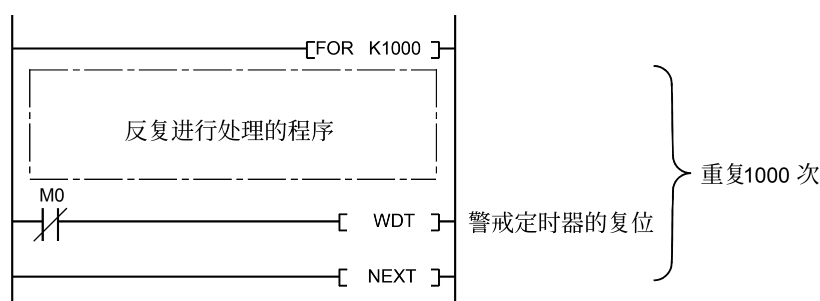
① 基本型QCPU正常动作，在警戒定时器的设置值以内执行END/FEND指令时，警戒定时器就不会达到时间。

② 基本型QCPU的硬件发生异常或因中断程序的执行而延长顺控程序的扫描时间，警戒定时器无法在设置值以内执行END/FEND指令的情况下，警戒定时器将达到时间。

（3） 注意事项

（a）警戒定时器的计测时间可产生0～10 ms范围内的误差。
设置警戒定时器的情况下，请考虑上述误差值后再行设置。

（b）警戒定时器可以通过顺控程序中执行WDT指令进行复位。
采用FOR指令和NEXT指令反复执行程序的情况下，出现警戒定时器达到时间时，利用WDT指令将警戒定时器复位。



（c） 即使顺控程序中对警戒定时器进行复位，扫描时间的值也不会复位。
扫描时间就是执行到END指令为止的过程中计测所得的时间。

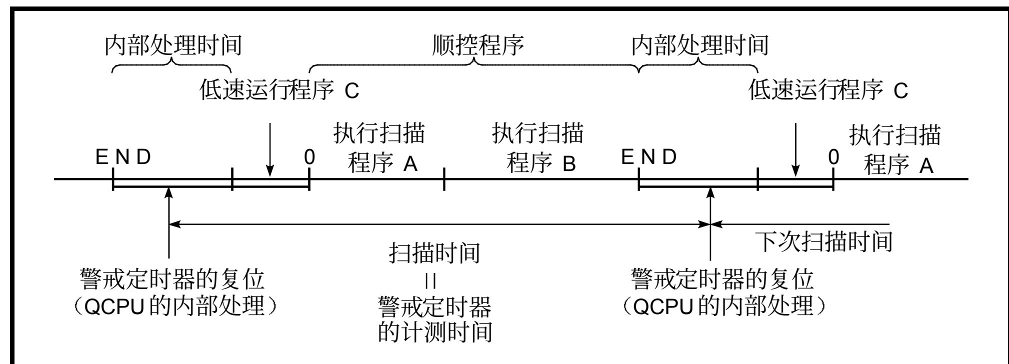


图7.8 警戒定时器的复位

备　注

- 扫描时间就是基本型QCPU的顺控程序从第0步开始执行到再次执行第0步为止的时间。
  每次扫描的扫描时间并不相同，它依据下列情况的不同而异。
  - 所使用的指令的执行/不执行
  - 中断程序的执行/不执行
- 希望每次扫描都用相同的扫描时间执行时，请使用恒定扫描功能。
  恒定扫描功能的详细内容请参照7.2节。

7.12 自我诊断功能

（1）什么是自我诊断功能

（a）自我诊断功能就是由基本型QCPU自身进行有无异常诊断的功能。

（b）自我诊断功能在防止基本型QCPU误动作的同时，还以预防故障为目的。
基本型QCPU的电源打开时或基本型QCPU的运行中发生异常的情况下，通过自我诊断功能检测出异常，显示出错，并停止基本型QCPU的运算。

（2）检测出异常时的处理

（a）基本型QCPU检测出异常的情况下，进行点亮ERR. LED等的处理。
此外，检测出异常时特殊继电器（SM0、SM1）接通，异常内容（出错代码）被存入特殊寄存器（SD0）　。
检测出多个异常时，将最新的出错的出错代码存入SD0。
请在程序中使用特殊继电器 、特殊寄存器将可编程控制器和机械类连锁。

（b）基本型QCPU记录最新的16个出错代码。（参照7.16节）
通过GX Developer的PC诊断模式，可以确认故障的记录。
即使断开电源，故障记录仍可通过电池备份。

（3）检测出异常时的基本型QCPU的动作

（a）通过自我诊断检测出异常的情况下，基本型QCPU的动作有以下所示的2种模式。

① 停止基本型QCPU运算的模式
检测出异常后，就在检出之时停止运算，并将输出（Y）全部设为断。

② 继续基本型QCPU运算的模式
检测出异常后，只有发生异常的程序（指令）不再执行，其他程序继续执行。

（b）发生以下的出错时可以通过PC参数的PC RAS设置选择运算的“继续/停止”。
（参数的默认值全部设置为“停止”。）
① 运算出错
② 扩展指令出错
③ 保险丝断路
④ 输入/输出模块对照出错
⑤ 智能程序执行出错
例如，输入/输出模块对照出错设置为“继续”的情况下，发生出错后，采用出错前的输入/输出地址继续运算。

（4）出错检查的选择

以下的出错检查可以通过PC参数的PC RAS设置选择“进行/不进行”。
（参数的默认值设置为全部“进行”。）
（a）电池检查
（b）保险丝断路检查
（c）输入/输出模块核对

自我诊断一览

诊断内容		出错信息	诊断时间
硬件异常	CPU异常	MAIN CPU DOWN	• 平时
	END指令不执行	END NOT EXECUTE	• 结束处理执行时
	RAM检查	RAM ERROR	• 电源接通及复位时
	运算电路检查	OPE.CIRCUIT ERR.	• 电源接通及复位时
	保险丝断路 （默认值…停止）∗1	FUSE BRAKE OFF	• END指令执行时 （默认值…检查）∗2
	输入/输出中断出错	I/O INT ERROR	• 中断发生时
	智能功能模块出错	SP.UNIT DOWN	• 电源接通及复位时 • FROM/TO指令执行时
	控制总线出错	CONTROL-BUS ERROR.	• 电源接通及复位时 • END指令执行时 • FROM/TO指令执行时
	发生瞬间掉电	AC/DC DOWN	• 平时
	电池低电压	BATTERY ERROR	• 平时 （默认值…检查）∗2
	输入/输出模块对照 （默认值…停止）∗1	UNIT VERIFY ERROR	• END指令执行时 （默认值…检查）∗2
操作异常	智能功能模块出错	SP.UNIT ERROR	• 指令执行时
	智能功能模块分配出错	SP.UNIT LAY ERR.	• 电源接通及复位时 • 从停止转换到运行时
	无参数	MISSING PARA.	• 电源接通及复位时
	引导出错	BOOT ERROR	• 电源接通及复位时
	指令无法执行	CAN'T EXE.PRG.	• 电源接通及复位时

∗1：采用GX Developer进行参数设置时可以更改为“继续”。
∗2：采用GX Developer进行参数设置时可以设置为“不检查”。

自我诊断一览（续）

诊断内容		出错信息	诊断时间
参数异常	参数设置检查	PARAMETER ERROR	• 电源接通及复位时 • 从停止转换到运行时
	通信参数出错	LINK PARA.ERROR	• 电源接通及复位时 • 从停止转换到运行时
	智能参数出错	SP.PARA.ERROR	• 电源接通及复位时
程序异常	指令代码检查	INSTRUCT CODE.ERR.	• 电源接通及复位时 • 从停止转换到运行时
	无END指令	MISSING END INS.	• 电源接通及复位时 • 从停止转换到运行时
	指针设置出错	CAN'T SET（P）	• 电源接通及复位时 • 从停止转换到运行时
	指针设置出错	CAN'T SET（I）	• 电源接通及复位时 • 从停止转换到运行时
	运算检查出错 （默认值…停止）＊1	OPERATION ERROR	执行指令时
	FOR～NEXT指令结构出错	FOR NEXT ERROR	执行指令时
	CALL～RET指令结构出错	CAN'T EXECUTE（P）	执行指令时
	中断程序出错	CAN'T EXECUTE（I）	执行指令时
CPU异常	运算迟缓监视	WDT ERROR	平时
	程序超时	PRG.TIME OVER	平时
信号报警器检查		F****	执行指令时

＊1：采用GX Developer进行参数设置时可以更改为“继续”。

7.12.1 发生出错时的LED显示

发生出错时，基本型QCPU正面上的ERR. LED将点亮/闪烁。
ERR. LED的动作的细节请参照7.19节。

7.12.2 出错的解除

只要是继续程序运算的出错，基本型QCPU就可以用程序进行解除操作。

（1）出错的解除

（a）出错解除步骤

出错的解除按以下步骤进行。

① 消除出错的原因。
② 将所解除的出错代码存入特殊寄存器。
③ 将特殊继电器SM50由断→通。
④ 出错被解除。

（b）出错解除后的状态

出错解除，CPU恢复后，与出错有关的特殊继电器、特殊寄存器以及ERR. LED恢复出错发生前的状态。
解除出错后再次发生同样的出错时，再次注册在故障记录中。

（c）信号报警器的解除

检测出多个出错时，信号报警器只解除最早检测出的F编号。

要　　点
将所解除的出错代码存入SD50解除出错时，后1位的代码将被忽略。

7.13 故障记录

基本型QCPU可以在利用自我诊断功能所检测出的结果上附加检测日期，作为故障记录存入存储器。

要　　点
故障检出时刻采用基本型QCPU内置时钟的时刻，因此，使用基本型QCPU时，请务必在开始时设置时间。

（1）存储区域
最新故障16点的量存储在经锁存的基本型QCPU的故障记录存储器中。

（2）存储数据
可编程控制器的电源接通期间，即使同一故障多次发生，也只在故障记录存储器中存入1次。

（3）故障记录的清零方法
故障记录存储器的清零利用GX开发器的PC诊断模式下的PC菜单中的故障记录清零进行。
一旦进行故障记录清零，基本型QCPU的故障记录存储器中的数据就被全部清除。

7.14 系统保护

基本型QCPU备有保护功能（系统保护），防止除设计者以外的第3者通过GX开发器、串行通信模块等对程序进行更改。

保护对象	保护有效的文件	保护内容	方法	有效时间	备注
文件单位	程序 软元件注释	以文件为单位按如下方法更改属性。 ①读写保护 ②写保护	利用口令注册更改文件的属性。	平时	

※在上述表中，控制指示、读写以及写入的意义如下。

项　　目	内　　容
读写	程序的读出、写入等的操作。
写入	程序的写入、测试等涉及写入处理的操作。

7.14.1 口令注册

口令用于禁止通过GX开发器读出及改写基本型QCPU内的程序、注释等的数据。

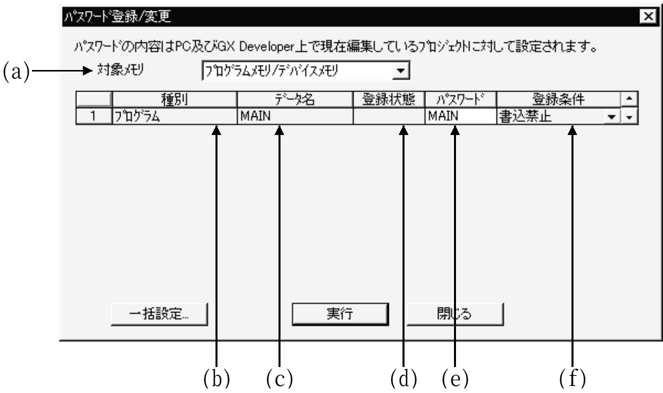
口令注册以程序存储器中的程序文件、软元件注释为对象。

注册的内容有以下2种。
- 文件名不可读出/写入
- 文件不可写入。（读出则可）

口令已注册的情况下，只要不输入相同的口令，就不能从GX开发器上进行文件的操作。

（1）口令的注册

口令的注册利用GX Developer的在线口令注册/PC写入的口令设置进行。



各项目的内容说明如下。

（a）对象存储器………………指定注册口令的文件所存储的存储器。

（b）类别………………………显示对象存储器中所存储的文件的类别。

（c）数据名……………………显示对象存储器中所存储的文件名。

（d）注册状态…………………口令已注册时显示“＊”。

（e）口令………………………设置初次注册的口令或当前已设置的口令。
口令设置后，就可以设置注册条件了。

（f）注册条件

① 写保护………………………禁止已指定口令的文件的写入。（读出则可。）

② 读/写保护…………………禁止已指定口令的文件的读出/写入。

③ 取消 ………………………取消所设置的口令。
（设置口令栏内当前已设置的口令。）

要　点

（1）口令注册有效的文件只有程序文件和软元件注释文件。
除此以外的文件不可注册口令。

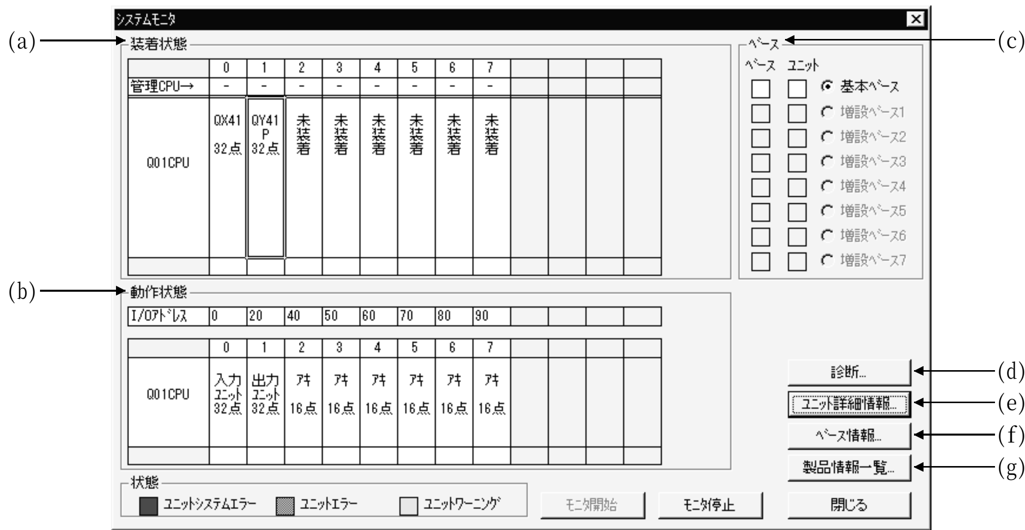
（2）文件注册的口令不可从文件中读出。
如果忘记了所注册的口令，就不能进行除以下内容以外的文件操作。
- 程序存储器：PC存储器格式化
- 标准ROM：程序存储器的ROM化

已注册的口令请记录在文书等上妥善保管。

7.15 采用GX Developer的系统监视

在基本型QCPU上连接个人电脑，通过GX Developer的系统监视（参照下图），可以进行，

- 安装状态
- 动作状态
- 模块的详细信息
- 产品信息的确认。



（a）安装状态

可以确认所选择的基板上安装的模块型号名称、点数。

未安装模块的插槽显示为“未安装”。

PC参数的输入/输出分配设置时设置为“空”的插槽即使安装了模块，也不显示模块的型号名称。

（b）动作状态

可以确认所选择的基板的各个插槽的输入输出编号、模块类别和点数。

动作状态为空0点，显示分配出错时，PC参数的输入/输出分配和实际安装状态不同。

请根据实际安装状态进行PC参数的输入/输出分配。

（c）基板

可以确认所使用的基板和所安装的模块的状态。

模块栏即使只有1个模块出现异常，也会显示状态。

（d）PC诊断

进行基本型QCPU的状态、出错的确认时使用。

（e）模块详细信息

用于确认所选择模块的详细信息的场合。

智能功能模块的详细信息的细节请参照各智能功能模块的手册。

（f）基板信息

基板信息项可以进行 “总体信息”和“基板信息” 的确认。

① 总体信息

可以确认所使用的基板数、基板上安装的模块数。

② 基板信息

可以确认所选择的基板（主基板、扩展基板1～7）的基板名、插槽数、基板类型、基板上安装的模块数。

（g）产品信息一览

可以确认所安装的CPU模块、输入输出模块、智能功能模块的个体信息（类别・系列・型号名称・点数・起始输入/输出・序列号・功能版本）。

序列号　功能版本

製品情報一覧

スロット	種別	シリーズ	形名	点数	先頭I/O	管理CPU	シリアルNo	Ver
CPU	CPU	Q	Q01CPU	-	-	-	030510000000000	A
0-0	-	-	空き	-	-	-	-	-
0-1	-	-	空き	-	-	-	-	-
0-2	-	-	空き	-	-	-	-	-
0-3	-	-	空き	-	-	-	-	-
0-4	-	-	空き	-	-	-	-	-
0-5	-	-	空き	-	-	-	-	-
0-6	-	-	空き	-	-	-	-	-
0-7	-	-	空き	-	-	-	-	-

CSVファイル作成　閉じる

7.16 LED的显示

基本型QCPU的正面装有表示基本型QCPU状态的LED。
本节阐述各LED的有关显示内容。

（1） 各LED的显示内容

LED名称	表示内容
运行	表示基本型QCPU的动作状态。 点亮：运行/停止/复位开关处于“运行”，正在运行中。 熄灭：运行/停止/复位开关处于“停止”，正在停止中。 或停止运行，检测出出错时。 闪烁：停止期间写入参数/程序，运行/停止/复位开关从“停止”→“运行”时。 程序写入后RUN LED亮起的情况下，进行以下的操作。 • 将运行/停止/复位开关从“运行” →“停止”→“运行”。 • 利用运行/停止/复位开关复位。 • 重新开启可编程控制器的电源。 参数写入后RUN LED亮起的情况下，进行以下的操作。 • 利用运行/停止/复位开关复位。 • 重新开启可编程控制器的电源。 （参数更改后将运行/停止/复位开关从“运行” →“停止”→“运行”的情况下，网络参数等与智能功能模块相关的参数上没有反映。）
ERR.	表示基本型QCPU的出错检测状态。 点亮：检测出不停止运行的自我诊断出错时。 （参数模式的PC RAS设置时将出错时的运行模式设置为“继续”） 用SET/OUT指令接通信号报警器。 熄灭：正常 闪烁：检测出停止运行的出错时。 利用运行/停止/复位开关复位时。（复位完毕后熄灭）
POWER	表示基本型QCPU内置电源的DV5V的输出状态。 点亮：DC5V输出正常 熄灭：可编程控制器的电源切断或DC5V输出异常。

（2）ERR. LED的熄灭方法

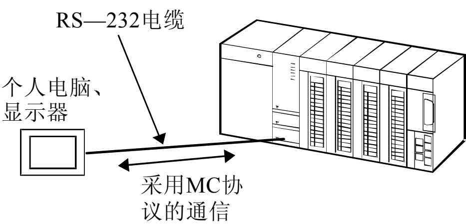
点亮的ERR. LED在出错的原因消除后，操作特殊继电器SM50、特殊寄存器SD50解除出错后熄灭。
（复位操作除外。）

备　　注

出错解除的有关内容请参照7.12.2项。

7.17 串行通信功能（可用于Q00CPU、Q01CPU上）

串行通信功能就是将CPU模块的RS—232接口和个人电脑、显示器等通过RS—232电缆连接，采用MC协议（＊1）实现通信的功能。
GX Developer、GX Configurator和CPU模块的连接不采用串行通信功能。
采用串行通信功能的通信可以在Q00CPU、Q01CPU上执行。
（Q00JCPU没有串行通信功能。）
本节将阐述使用串行通信功能和个人电脑、显示器等进行通信时必要的技术规范、功能和各种设置。



＊1：所谓MC协议，就是MELSEC通信协议的简称。
MELSEC通信协议是依照Q系列可编程控制器（串行通信模块、Ethernet接口等）的通信步骤，用于从对方的设备向QCPU存取数据的通信方式的名称。
具体通信方式2种，即采用ASCII码数据的通信方式和采用2进制码数据的通信方式。

要　　点
可以采用串行通信功能从个人电脑、显示器等实现通信的只有和个人电脑、显示器等相连接的Q00/Q01CPU。 不可通过和个人电脑、显示器等相连接的Q00/Q01CPU，和MELSECNET/H、Ethernet、CC—Link的其他站进行通信。

（1）技术规范

（a）传输规格

CPU模块的串行通信功能所使用的RS—232的传输规格如下表所示。

请确认个人电脑、显示器等的技术规范符合下表后，再使用串行通信功能。

项　目	默 认 值	设置范围
通信方式	全双工通信	———
同步方式	起止同步方式	———
传输速度　＊1	19.2kbps	9.6kbps、19.2kbps、38.4kbps、57.6kbps、115.2kbps
数据形式	起始位：1 数据位：8 校验位：奇数校验 停止位：1	———
MC协议格式　＊2 （自动判别）	格式4（ASCII） 格式5（2进制）	———
帧　＊2	QnA互换3C帧 QnA互换4C帧	———
传输控制	DTR/DSR控制	———
和数校验　＊1	无	有、无
传输等待时间　＊1	无等待	无等待，10 ms～150 ms（10 ms单位）
运行种写入设置　＊1	不允许	允许，不允许
延长距离	15m	———

＊1：可以在GX Developer的PC设置时进行设置。

＊2：MC协议形式和结构之间的关系如下表所示。

功能		格式4	格式5
采用ASCII码的通信	QnA互换3C帧	○	×
	QnA互换4C帧	○	×
采用2进制码的通信	QnA互换4C帧	×	○

○：可以使用，×：不可使用

（b）RS—232接插件的规格

Q00/Q01CPU的RS—232接插件的用途如下表所示。

外观	针号	信号名称	信号名称
MiniDin（6针）（雌接头）	1	RD（RXD）	数据接收
	2	SD（TXD）	数据发送
	3	SG	信号地
	4	———	———
	5	DSR（DR）	数据集就绪
	6	DTR（ER）	数据终端就绪

（c）RS—232电缆

个人电脑/显示器等和Q00/Q01CPU的连接可采用以下的RS—232电缆。

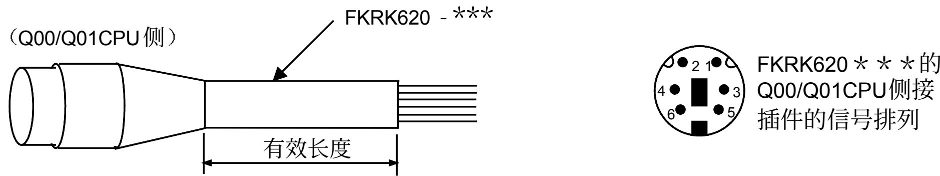
- QC30R2（电缆长度：3 m）
- FKRK620—＊＊＊（仓茂电工株式会社出品）

一侧：带微型接插件；另一侧：无接插件的电缆

＊＊＊表示电缆长度，可以以0.1m位单位最长指定为15m

联络处：仓茂电工株式会社　名古屋支店

TEL：052-332-2781　　FAX：052-331-2430



针号	1	2	3	4	5	6	金属壳
信号名称	RD	SD	SG	—	DR	ER	
芯线	黄	茶	黑	红	蓝	绿	屏蔽

（2）功能

串行通信功能中可以执行下表的MC协议指令。

有关MC协议的详细内容请参照下列手册。

• Q对应MELSEC通信协议参考手册

功　能			指令	处理内容	处理点数
元件寄存器	一次性成批读出	位单位	0401（00□1）	以1点为单位读出位软元件。	ASCII ：3584点 BIN ：7168点
		字单位	0401（00□0）	以16点为单位读出位软元件。	480字（7680点）
				以1点为单位读出字软元件。	480点
	一次性成批写入 ＊1	位单位	1401（00□1）	以1点为单位写入位软元件。	ASCII：3584点 BIN：7168点
		字单位	1401（00□0）	以16点为单位写入位软元件。	480字（7680点）
				以1点为单位写入位软元件。	480点
	随机读出	字单位	0403（00□0）	以16点、32点为单位，随机指定元件•元件号读出位软元件。	96点
				以1点、2点为单位，随机指定元件•元件号读出字软元件。	
	测试 ＊1 （随机写入）	位单位	1402（00□1）	以1点为单位，随机指定元件•元件号，对位软元件进行设置、清零。	94点
		字单位	1402（00□0）	以16点、32点为单位，随机指定元件•元件号，对字软元件进行设置、清零。	＊2
	监视注册	字单位	0801（00□0）	以16点、32点为单位注册所监视的位软元件。	96点
				以1点、2点为单位注册所监视的位软元件。	96点
	监视	字单位	0802（00□0）	对经过监视注册的元件进行监视。	监视注册点数

＊1：CPU模块进行运行中写入的情况下，运行中写入设置须设置为“允许”。

＊2：处理点数设置在下式的范围内。

（字存取点数）×12＋（双字存取点数）×14≤960

• 位软元件在字存取时1点为16位；双字存取时1点为32位。

• 字软元件在字存取时1点为1个字；双字存取时1点为2个字。

（3） 可以存取的元件

<table>
<tr><th>分 类</th><th colspan="2">设 备</th><th>元件代码元件代码</th><th>元件号范围∗1
（默认值）</th><th>写入</th><th>读出</th></tr>
<tr><td rowspan="5">内部系统
软元件</td><td colspan="2">功能输入</td><td>FX</td><td>000000～00000F</td><td rowspan="3">×</td><td rowspan="3">×</td></tr>
<tr><td colspan="2">功能输出</td><td>FY</td><td>000000～00000F</td></tr>
<tr><td colspan="2">功能寄存器</td><td>FD</td><td>000000～000004</td></tr>
<tr><td colspan="2">特殊继电器</td><td>SM</td><td>000000～001023</td><td rowspan="22">○</td><td rowspan="28">○</td></tr>
<tr><td colspan="2">特殊寄存器</td><td>SD</td><td>000000～001023</td></tr>
<tr><td rowspan="26">内部用户
软元件</td><td colspan="2">输入</td><td>X</td><td>000000～0007FF</td></tr>
<tr><td colspan="2">输出</td><td>Y</td><td>000000～0007FF</td></tr>
<tr><td colspan="2">内部继电器</td><td>M</td><td>000000～008191</td></tr>
<tr><td colspan="2">锁存继电器</td><td>L</td><td>000000～002047</td></tr>
<tr><td colspan="2">信号警报器</td><td>F</td><td>000000～001023</td></tr>
<tr><td colspan="2">边沿继电器</td><td>V</td><td>000000～001023</td></tr>
<tr><td colspan="2">通信继电器</td><td>B</td><td>000000～0007FF</td></tr>
<tr><td colspan="2">数据寄存器</td><td>D</td><td>000000～011135</td></tr>
<tr><td colspan="2">通信寄存器</td><td>W</td><td>000000～0007FF</td></tr>
<tr><td rowspan="3">定时器</td><td>触点</td><td>TS</td><td rowspan="3">000000～000511</td></tr>
<tr><td>线圈</td><td>TC</td></tr>
<tr><td>当前值</td><td>TN</td></tr>
<tr><td rowspan="3">累加
定时器</td><td>触点</td><td>SS</td><td rowspan="3">———</td></tr>
<tr><td>线圈</td><td>SC</td></tr>
<tr><td>当前值</td><td>SN</td></tr>
<tr><td rowspan="3">计数器</td><td>触点</td><td>CS</td><td rowspan="3">000000～000511</td></tr>
<tr><td>线圈</td><td>CC</td></tr>
<tr><td>当前值</td><td>CN</td></tr>
<tr><td colspan="2">特殊通信继电器</td><td>SB</td><td>000000～0003FF</td></tr>
<tr><td colspan="2">特殊通信寄存器</td><td>SW</td><td>000000～0003FF</td></tr>
<tr><td colspan="2">步进继电器 ∗2</td><td>S</td><td>000000～002047</td><td>×</td></tr>
<tr><td colspan="2">直接输入</td><td>DX</td><td>000000～0007FF</td><td rowspan="5">○</td></tr>
<tr><td colspan="2">直接输出</td><td>DY</td><td>000000～0007FF</td></tr>
<tr><td colspan="2">变址寄存器</td><td>Z</td><td>000000～000009</td></tr>
<tr><td colspan="2" rowspan="2">文件寄存器</td><td>R</td><td>000000～032767</td></tr>
<tr><td>ZR</td><td>000000～007FFF</td></tr>
</table>

○：可读出/写入，×：不可写入

∗1：上表所示的软元件号的范围为默认值。
Q00/Q01CPU上更改软元件点数的情况下，请在更改后的软元件号范围内使用。

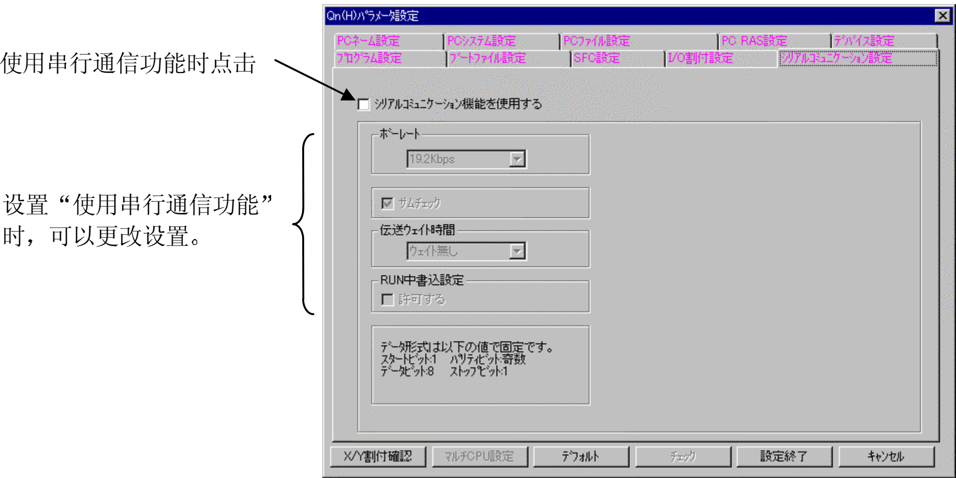
∗2：基本型QCPU不对应SFC功能，因此，即使对步进继电器进行读出，也不可作为数据使用。

（4）传输规格的设置

串行通信功能的传输速度、和数校验、传输等待时间、运行中写入设置在PC参数的串行通信设置时进行设置。

（a）利用串行通信功能和个人电脑、显示器等通信的情况下，设置“使用串行通信功能”。

（b）显示传输速度、和数校验、传输等待时间、运行中写入设置的默认值。
可以根据外部设备的性能更改传输速度、和数校验、传输等待时间、运行中写入的设置。



（5）注意事项

（a）利用串行通信功能和个人电脑、显示器等通信期间，可以将连接转换到GX Developer上。
但是，利用串行通信功能进行通信的个人电脑、显示器等将出现通信出错。
重新将个人电脑、显示器等和CPU模块连接起来时的个人电脑、显示器等的启动方法的有关内容请参照所使用的设备的手册。

（b）设置“使用串行通信功能”时，即使GX Developer上更改传输速度，也不会有效。

要　　点
串行通信设置时所设置的数据在以下情况下有效。 • 可编程控制器的电源接通时 • Q00/Q01CPU复位时

（6）利用串行通信功能通信时的出错代码

利用串行通信功能通信期间发生出错时，从Q00/Q01CPU向外部元件传送的出错代码、出错内容及其处理如下表所示。

出错代码（十六进制）	出错项目	出错内容	处置方法
4000H ～ 4FFFH	——	（CPU所检测出的出错） *发生的非串行通信功能出错	• 参照基本型QCPU（Q模块）用户手册（硬件设计 • 保养 • 检查篇）的附录进行处置。
7153H	帧长度出错	• 接收的电文长度超过允许范围。	• 重新评估所发送的电文。 • 在存取点数的允许范围内构建电文。
7155H	监视未注册出错	• 监视注册前发出监视请求。	• 事先注册希望监视的软元件后再请求监视。
7164H	请求内容出错	• 请求内容或软元件指定方法存在出错。	• 确认对方设备的传送电文/请求内容，修正后重新通信。
7167H	运行中不可	• 运行中写入设置为否时指定了写入指令。	• 将设置更改为运行中写入可后重新通信。
7168H		• 指定了运行中无法执行的指令。	• 将CPU模块停止后重新通信。
716DH	监视注册出错	• 未用QnA互换3C/4C结构进行监视注册。	• 重新进行监视注册
7E40H	指令出错	• 指定了不存在的指令或子指令。	• 确认对方设备的传送电文，修正后重新通信。
7E41H	数据长度出错	• 指定了超出随机读出/写入时可以通信的点数。	• 确认对方设备的传送电文，修正后重新通信。
7E42H	数据数出错	• 请求点数对指令而言已超出范围	• 确认对方设备的传送电文，修正后重新通信。
7E43H	软元件出错	• 指定了不存在的软元件。 • 指定了该指令不能指定的软元件。	• 确认对方设备的传送电文，修正后重新通信。
7E47H	连续请求出错	• 响应电文返回前接收了下一个请求。	• 不从对方设备上连续发出请求。 • 将定时器1的监视时间和对方设备一侧的定时器到达时间对准。
7E4FH	软元件点数出错	• 存取点数不正确。	• 确认对方设备的传送电文，修正后重新通信。
7E5FH	被请求模块输入/输出编号出错	• 被请求模块的输入/输出编号出错。	• 修正数据接收方的输入/输出编号。
7E64H	注册点数范围出错	• 注册点数（字/位）超出范围。	• 修正注册点数（字/位）的设置值。
7F01H	缓冲器出错	• 接收数据的处理完成前又接收了下一个数据。	• 采用和对方设备交换信号等方法拉开时间间隔。
7F21H	接收信息头部错误	• 命令（帧）部分存在指定出错。 • 接收了无法变换为二进制数的ASCII码	• 确认对方设备的传送电文，修正后重新通信。

出错代码（十六进制）	出错项目	出错内容	处置方法
7F22H	命令出错	• 指定了不存在的指令或软元件。 • 远程口令长度出错。	• 确认对方设备的传送电文，修正后重新通信。
7F23H	MC协议电文出错	• 字符部分之后无数据（ETX、CR—LF等），或指定出错。	• 确认对方设备的传送电文，修正后重新通信。
7F24H	和数校验出错	• 计算出的和数校验和所接收的和数校验不符合。	• 重新评估对方设备的和数校验。
7F67H	超限出错	• Q00CPU、Q01CPU接收处理完成前接收了下一个数据。	• 降低通信速度，重新通信。 • 确认Q00CPU、Q01CPU有未发生瞬间掉电。（Q00CPU、Q01CPU的情况下，可以采用特殊寄存器SD53进行确认） 发生瞬间掉电时，消除其原因。
7F69H	奇偶性出错	• 奇偶位的设置不相符合。	• 使Q00CPU、Q01CPU和对方设备的设置一致。
7F6AH	缓冲溢出出错	• OS的接收缓冲溢出，接收数据丢失。	• 进行DTR控制，在缓冲不溢出的前提下进行通信。
F000H	———	• MELSECNET/H网络所检测出的出错。	• 参照Q用MELSECNET/H参考手册（PC间网络篇）进行处置。

第8章　与智能功能模块之间的通信

(1) 什么是智能功能模块
基本型QCPU上可以使用Q系列对应的智能功能模块。
智能功能模块就是用于将输入输出模块无法处理的模拟量、高频脉冲交给基本型QCPU处理的模块。
例如，模拟量可以利用智能功能模块的模/数转换模块转换为数字量后使用。

(2) 智能功能模块的数据读出/写入
智能功能模块备有存储器（缓冲存储器），用于存储从外部接收的数据以及准备向外部输出的数据。
基本型QCPU对智能功能模块的缓冲存储器进行数据的读出/写入。

8.1　基本型QCPU和Q系列对应智能功能模块之间的通信

基本型QCPU可以采用以下方法和智能功能模块进行通信。
- 采用GX Configurator的初始设置、自动刷新设置
- 智能功能模块元件（Y□￥G□）
- 智能功能模块专用指令
- FROM/TO指令

上述和智能功能模块的通信方法的通信时机如下表所示。

和智能功能模块的通信方法		通信时机					存储位置　*1	
		电源接通	基本型QCPU复位	停止→运行	指令执行	结束处理	*2 基本型QCPU	*3 智能
GX Configurator	初始设置	○	○	○	−	−	○	−
	自动刷新设置	−	−	−	−	○	○	−
智能功能模块软元件　*4		−	−	−	○	−	○	−
智能功能模块专用指令　*4		−	−	−	○	−	○	−
FROM/TO指令　*4		−	−	−	○	−	○	−

通信时机……○：执行，−：不执行
存储位置……○：可以存储，−：不可存储

备　注

*1：表示用GX Configurator设置的数据是存储在基本型QCPU上还是智能功能模块一侧。
*2：表示基本型QCPU的内置存储器。
*3：表示智能功能模块。
*4：表示使用智能功能模块软元件、智能功能模块专用指令、FROM/TO指令的程序。

8

8.1.1 采用GX Configurator进行初始设置、自动刷新设置

(1) 智能功能模块的初始设置、自动刷新设置
通过在GX Developer上增加智能功能模块对应的GX Configurator，可以由GX Developer起动GX Configurator进行初始设置和自动刷新设置。
智能功能模块的初始设置、自动刷新设置一经设置，即使不编写和智能功能模块通信的程序，也可以进行数据的写入/读出。
而且，即使不指定智能功能模块的缓冲存储器地址，也可以进行初始设置和自动刷新设置。

(2) 采用GX Configurator的设置
以下以A/D（模/数）转换模块Q64AD的初始设置、自动刷新设置时的情况为例进行说明。

(a) 初始设置

① Q64AD的初始设置有如下所示的3种。
- A/D转换允许/禁止设置
- 采样/平均处理指定
- 时间平均/次数平均指定
- 平均时间/平均次数指定

② Q64AD的初始设置在下图所示的GX Configurator初始设置画面上进行。

【初始设置画面】

初期設定

ﾕﾆｯﾄ情報
ﾕﾆｯﾄ形名： Q64AD　　先頭 I/O No. 10
ﾕﾆｯﾄ種別： A/D変換ﾕﾆｯﾄ

設定項目	設定
CH.1 A/D変換許可/禁止設定	許可
CH.1 ｻﾝﾌﾟﾘﾝｸﾞ/平均処理指定	ｻﾝﾌﾟﾘﾝｸﾞ
CH.1 時間平均/回数平均指定	回数平均
CH.1 平均時間/平均回数設定 入力範囲:時間2～5000(ms)/回数4～62500(回)	0
CH.2 A/D変換許可/禁止設定	許可
CH.2 ｻﾝﾌﾟﾘﾝｸﾞ/平均処理指定	ｻﾝﾌﾟﾘﾝｸﾞ
CH.2 時間平均/回数平均指定	回数平均
CH.2 平均時間/平均回数設定 入力範囲:時間2～5000(ms)/回数4～62500(回)	0
CH.3 A/D変換許可/禁止設定	許可

各種情報
選択入力

ﾃｷｽﾄﾌｧｲﾙ作成　設定終了　ｷｬﾝｾﾙ

所设置的初始设置数据存入智能功能模块。

(b) 自动刷新设置

自动刷新设置中，对存储如下所示数据的基本型QCPU一侧的软元件进行设置。

- Q64AD的默认输出
- Q64AD的最大值/最小值
- 出错代码

Q64AD的自动刷新设置在下图所示的GX Configurator的自动刷新画面上进行。

【自动刷新设置画面】

自動ﾘﾌﾚｯｼｭ設定

ﾕﾆｯﾄ情報

ﾕﾆｯﾄ形名： Q64AD　　先頭 I/O No.　10

ﾕﾆｯﾄ種別： A/D変換ﾕﾆｯﾄ

	ﾕﾆｯﾄ側				CPU側
設定項目	ﾊﾞｯﾌｧｻｲｽﾞ	転送ﾜｰﾄﾞ数	ﾊﾞｯﾌｧｵﾌｾｯﾄ	転送方向	ﾃﾞﾊﾞｲｽ
CH.1 ﾃﾞｨｼﾞﾀﾙ出力値	1	1	0	->	
CH.2 ﾃﾞｨｼﾞﾀﾙ出力値	1	1	0	->	
CH.3 ﾃﾞｨｼﾞﾀﾙ出力値	1	1	0	->	
CH.4 ﾃﾞｨｼﾞﾀﾙ出力値	1	1	0	->	
CH.1 最大値	1	1	0	->	
CH.1 最小値	1	1	0	->	
CH.2 最大値	1	1	0	->	
CH.2 最小値	1	1	0	->	
CH.3 最大値	1	1	0	->	
CH.3 最小値	1	1	0	->	
CH.4 最大値	1	1	0	->	

ﾃｷｽﾄﾌｧｲﾙ作成　　設定終了　　ｷｬﾝｾﾙ

所设置的自动刷新设置数据存入基本型QCPU的智能参数。

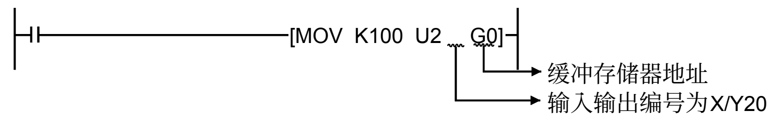
备　注

GX Configurator的详细内容请参照所使用的智能功能模块的手册。

8.2 采用智能功能模块软元件（U□￥G□）的通信

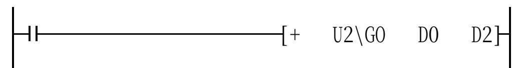
(1) 智能功能模块软元件

智能功能模块软元件就是将智能功能模块的缓冲存储器以基本型QCPU的软元件形式加以表达。
可以读出智能功能模块的缓冲存储器中存储的数据，或向智能功能模块的缓冲存储器写入数据。
例如，向输入输出编号为XY/Y20～X/Y2F的智能功能模块的缓冲存储器地址0中写入“100”时，可以采用如下形式编程。



(2) 与FROM/TO指令的不同点

由于智能功能模块软元件可以当作基本型QCPU的软元件操作，因此，从智能功能模块读出的数据的加工可以用1条指令进行。
例如，将从智能功能模块接收的数据和D0的数据相加，存入D2时的编程方法如下。



由此可以减少程序的总体步数。
而处理速度就是指令的执行时间和智能功能模块存取时间的合计值。

要　　点

程序中繁复地读出智能功能模块的数据时，请采用每次通过智能功能模块软元件的使用，使用FROM指令读出程序的1处，再存入数据寄存器等中的方法。
由于智能功能模块软元件是每次执行指令时与智能功能模块进行存取的，因此，程序的扫描时间不会延长。

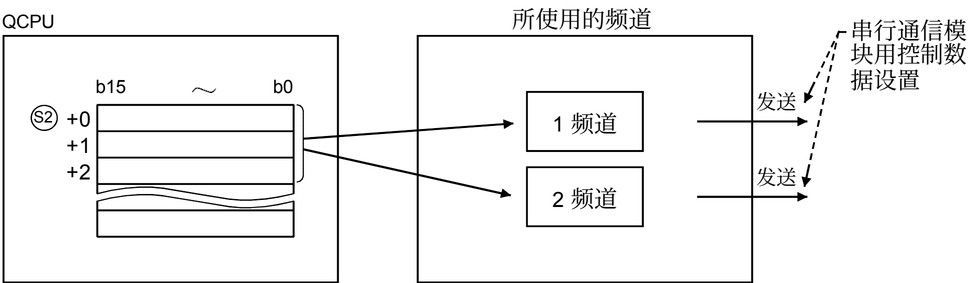
备　　注

智能功能模块元件的有关内容请参照10.5节。

8.3 采用智能功能模块专用指令的通信

(1) 什么是智能功能模块专用指令

(a) 智能功能模块专用指令就是将使用智能功能模块功能的程序简化的指令。
例如，使用串行通信模块专用的OUTPUT指令，可以利用无顺序协议，以任意的电文格式传输数据。
此时，可以不必考虑串行通信模块的缓冲存储器地址而与对方设备进行通信。



(b) 智能功能模块专用指令
智能功能模块专用指令中可以指定完成软元件。
该完成软元件在智能功能模块专用指令执行完毕后的1次扫描中接通。
完成软元件接通的情况下，不可在同一智能功能模块上执行智能功能模块专用指令。
同一智能功能模块上要使用多条智能功能模块专用指令时，请在完成软元件接通时执行下一条智能功能模块专用指令。

(2) 注意事项
执行智能功能模块专用指令，完成软元件接通之前就将基本型QCOU从运行设为停止，则下次运行的1次扫描完毕后软元件接通。

备　注

智能功能模块专用指令、完成软元件的有关内容请参照所使用智能功能模块的手册。

8.4 采用FROM/TO指令的通信

(1) FROM/TO指令

FROM/TO指令执行时，可以读出智能功能模块的缓冲存储器中存储的数据，或向智能功能模块的缓冲存储器写入。

FROM指令将从智能功能模块的缓冲存储器中读出的数据存入指定软元件。

TO指令将指定软元件的数据写入智能功能模块的缓冲存储器。

备　注

1) FROM/TO指令的有关详情请参照下列手册。
   - QCPU（Q模块）/QnACPU编程手册（通用指令篇）

2) 智能功能模块的缓冲存储器的有关详情请参照所使用的智能功能模块的手册。

第9章　参数

基本型QCPU的参数有2种，即单独使用可编程控制器的系统时设置的“PC参数”和使用MELSECNET/H、Ethernet、CC—Link时设置的“网络参数”。

本章将采用GX Developer设置的PC参数和网络参数整理成一览表。
各设置项目的详细内容请参照一览表中记载的参照项或参照手册。
采用GX Developer进行设置操作的有关内容请参照GX Developer的操作手册。

以下情况下，从GX Developer写入的参数（＊1）在基本型QCPU的内部有效。
- 可编程控制器的电源接通时
- 基本型QCPU复位时
- 基本型QCPU从停止→运行时

＊1：PC参数的输入/输出分配的开关设置、详细设置以及网络参数在如下所述的基本型QCPU的启动时被传送到设置对象的智能功能模块并使用。
- 可编程控制器的电源接通时
- 基本型QCPU复位时

要　点

（1）PC参数的输入/输出分配的开关设置和网络参数发生变更的情况下，必须重新开启可编程控制器的电源（通→断→通）或将基本型QCPU复位。
如果不重新开启可编程控制器电源（通→断→通）或将基本型QCPU复位，所更改的PC参数的输入/输出分配的开关设置和网络参数将不会生效。
（2）PC参数写入基本型QCPU时，请重新开启可编程控制器电源（通→断→通）或将基本型QCPU复位。

备　注

基本型QCPU从停止→运行时，PC参数的输入/输出分配的开关设置、详细设置以及网络参数不传送到设置对象的智能功能模块中。

备忘录

9

表9.1 参数一览表

项　目			内　　容	
PC名设置			设置所使用的CPU模块的标贴、注释。 PC名设置时即使设置标贴、注释也不影响实际动作。	
	标贴		设置CPU模块的标贴（名称、用途）。	
	注释		设置CPU模块标贴的注释。	
PC系统设置			使用CPU模块的情况下必须进行设置。 也在可以保持默认值不变的情况下进行控制。	
	定时器 时限设置	低速	设置低速定时器/高速定时器的时限。	
		高速		
	运行—暂停触点		设置控制CPU模块的运行/暂停的触点。 不可只设置暂停触点。（运行触点、运行＋暂停触点的设置则可。）	
	远程复位		设置来自GX Developer的远程操作的允许/禁止。	
	停止—运行输出模式		设置从停止状态转换到运行状态时的输出（Y）状态。	
	空插槽数		设置主基板/扩展基板的空插槽数。	
	系统 中断设置	中断计数器 起始号	设置中断计数器的起始号。	
		I n固定周期间隔 （n：28～31）	设置中断指针（I28～31）的执行间隔。	
	中断程序/固定周期程序设置		设置中断程序是高速执行还是不执行。	
	模块同步设置		设置使CPU模块的启动和智能功能模块的启动同步还是不同步。	
PC RAS设置			设置RAS功能所需的各种设置。	
	WDT（警戒定时器） 设置	WDT设置	设置CPU模块的警戒定时器的时间。	
	出错时的运行模式		设置检测出出错时的CPU模块的动作模式。	
	出错检查		设置是检查指定出错还是不检查。	
	恒定扫描		设置恒定扫描的时间。	

	默认值	设置范围	参照
	－－－－	－－－－	－
	无设置	最多10个半角字符	－
	无设置	最多64个半角字符	－
	－－－－	－－－－	－
	100ms	1 ms～1000 ms（1 ms单位）	10.2.10项
	10.0ms	0.1 ms～100.0 ms（0.1 ms单位）	10.2.10项
	无设置	X0～X7FF	7.6.1项
	不允许	允许/不允许	7.6.3项
	输出停止前的输出（Y）	输出停止前的输出（Y）/输出清零（1次扫描后输出）	7.4节
	16点	Q00JCPU：16点/32点/64点/128点/256点 Q00CPU/Q01CPU：16点/32点/64点/128点/256点/512点/1024点	5.6.1项
	无设置	C0～C13408（计数器设置点数最多可设置至-128点。）	10.2.11项
	I28:100.0ms I29:40.0ms I30:20.0ms I31:10.0ms	2 ms～1000 ms（1 ms单位）	10.10节
	不高速执行	不高速执行/高速执行	4.1.3项
	与智能功能模块的启动同步	与智能功能模块的启动同步/不同步	－
	－－－－	－－－－	－
	200ms	10 ms～2000 ms（10 ms单位）	4.2.节 7.11节
	停止	停止/继续	7.13节
	进行检查	进行检查/不进行检查	7.13节
	无设置	1 ms～2000 ms（1 ms单位）	7.2节

表9.1 参数一览表（续）

项　目			内　　容
软元件设置			设置每个软元件使用点数、锁存范围、本地软元件范围。
	软元件数		根据系统设置软元件的使用数。
	锁存（1）起始/最终 （锁存区域有效）		设置可以采用远程锁存清零操作进行清零的锁存范围（起始软元件号/最终软元件号）。
	锁存（2）起始/最终 （锁存区域无效）		设置不可采用远程锁存清零操作进行清零的锁存范围（起始软元件号/最终软元件号）。
引导文件设置			设置是从标准ROM上引导还是不引导。
输入/输出分配			设置系统的各个模块的安装状态。
	输入/输出分配	类别	设置所安装的模块的类别。
		型号名称	设置所安装的模块的型号名称。 （CPU模块为不使用的用户的备望）
		点数	设置各插槽的点数。
		起始XY （起始输入输出编号）	设置各插槽的起始输入输出编号。
	基本设置	基板型号名称	设置所使用的主基板、扩展基板的型号名称。 （CPU模块为不使用的用户的备望）
		电源模块型号名称	设置主基板、扩展基板上安装的电源模块的型号名称。 （CPU模块为不使用的用户的备望）
		扩展电缆型号名称	设置扩展电缆的型号名称。（CPU模块为不使用的用户的备望）
		插槽数	设置主基板、扩展基板的插槽数。 插槽数的设置对所有基板进行。
	锁存设置		设置智能功能模块的各种开关。
	详细设置	出错时的 输出模式	设置智能功能模块出现停止出错时的清零/保持。
		H/W出错时的CPU动作模式	设置智能功能模块的硬件异长时停止/继续CPU模块运行。
		输入/输出响应时间	设置输入模块、输入输出混合模块、高速输入模块、中断模块的响应时间。

	默认值	设置范围	参照
	————	————	—
	X:2k点 Y:2k点 M:8k点 L:2k点 B:2k点 F:1k点 SB:1k点 V:1k点 S:2k点 T:512点 ST:0k点 C:512点 D:11136点 W:2k点 SW:1k点	X（2 k点）、Y（2 k点）、S（2 k点）、SB（1 k点）、SW（1 k点）为固定。 可在合计16.4 k字的范围的进行设置，包括上述点数（1.5k字）。 • 1个元件：最大32 k点 （位软元件的合计点数没有限制。）	10.1节 10.2节
	无设置	B,F,V,T,ST,C,D,W的各软元件只设置1个范围。	7.3节
	无设置	L,B,F,V,T,ST,C,D,W的各软元件只设置1个范围。	7.3节
	不进行引导	不进行引导，进行引导	6.5.1项
	————	————	—
	无设置	• 空、输入、高速输入、输出、智能、输入输出混合、中断	5.6节
	无设置	• 半角16字符	
	无设置	• Q00JCPU：16点/32点/64点/128点/256点 • Q00CPU/Q01CPU：16点/32点/64点/128点/256点/512点/1024点	
	无设置	• Q00JCPU：0H～F0H • Q00CPU./Q01CPU：0H～3F0H	
	无设置	• 半角16字符	5.3节
	无设置	• 半角16字符	
	无设置	• 半角16字符	
	无设置	• 2,3,5,8,10,12	
	无设置	• 参照所使用的智能功能模块的手册	7.8节
	清零	• 清零/保持	—
	停止	• 停止/继续	
	输入、输入输出混合：10 ms 高速输入、中断：0.2 ms	• 输入、输入输出混合：1 ms、5 ms、10 ms、20 ms、70 ms • 高速输入、中断：0.1 ms、0.2 ms、0.4 ms、0.6 ms、1 ms	7.7节

表9.1 参数一览表（续）

项　目	内　　容	
X/Y分配确认	可以对输入/输出分配、MELSECNET/Ethernet的设置，CC—Link设置中所设置的内容进行确认。	
串行通信设置	设置使用Q00/Q01CPU的串行通信功能时的传输速度、和数校验、传输等待时间、运行中写入可否。	
使用串行通信功能	使用串行通信功能时，附加检查标记。	
波特率	设置与对方元件进行数据通信时的通信速率。	
和数校验	设置利用串行通信功能进行数据通信时，根据对方设备的性能，设置在发送电文、接收电文上是否附加校验码。	
发送等待时间	设置对方设备发送数据后，无法立刻接收数据时的基本型QCPU的发送等待时间。	
运行中写入设置	设置对方设备向可编程控制器CPU写入数据时，即使可编程控制器CPU处于运行中，是否仍可写入。	
网络参数	设置MELSECNET、Ethernet、CC—Link所使用的参数。	
MELSECNET、Ethernet设置	设置MELSECNET、Ethernet的网络参数。	
CC—Link设置	设置CC—Link的参数。	
智能功能模块参数	设置采用GX Developer设置的智能功能模块的初始设置、自动刷新设置。	

	默认值	设置取范围	参照
	————	————	—
	————	————	—
	————	————	—
	19.2kbps	9.6kbps/19.2kbps/38.4kbps/57.6kbps/115.2kbps	7.17节
	有	无/有	
	无等待	无等待/10 ms～150ms（10 ms单位）	
	不允许	不允许/允许	
	————	————	—
	无设置	• 参照Q对应MELSECNET/H、Ethernet的手册	—
	无设置	• 参照CC—Link的手册	—
	无设置	参照所使用的智能功能模块的手册	—

第10章　软元件的说明

本章阐述基本型QCPU上可以使用的软元件。

10.1　软元件一览

基本型QCPU上可以使用的软元件名称和使用范围如表10.1所示。

表10.1软元件一览表

分类	类别	软元件名称	默认值		采用参数设置的设置范围	参照
			点数	使用范围		
内部用户软元件	位软元件	输入*3	2048点	X0～X7FF	可在16.4 k字以内更改*3	10.2.1项
		输出*3	2048点	Y0～Y7FF		10.2.2项
		内部继电器	8192点	M0～M8191		10.2.3项
		锁存继电器	2048点	L0～L2047		10.2.4项
		信号报警器	1024点	F0～F1023		10.2.5项
		边沿继电器	1024点	V0～V1023		10.2.6项
		步进继电器*3*4	2048点	—		10.2.9项
		通信用特殊继电器*3	1024点	SB0～SB3FF		10.2.8项
		通信继电器	2048点	B0～B7FF		10.2.7项
	字软元件	定时器*1	512点	T0～T511		10.2.10项
		保持定时器*1	0点	（ST0～ST511）		
		计数器*1	512点	C0～C511		10.2.11项
		数据寄存器	11136点	D0～D11135		10.2.12项
		通信寄存器	2048点	W0～W7FF		10.2.13项
		通信用特殊寄存器*3	1024点	SW0～SW3FF		10.2.14项
内部系统软元件	位软元件	功能输入	5点	FX0～FX4	不可	10.3.1项
		功能输出	5点	FY0～FY4		10.3.1项
		特殊继电器	1000点	SM0～SM999		10.3.2项
	字软元件	功能寄存器	5点	FD0～FD4		10.3.1项
		特殊寄存器	1000点	SD0～SD999		10.3.3项

要　　点

＊4：步进继电器是SFC用的软元件。
　　基本型QCPU不支持SFC程序，因此，不可使用步进继电器。

备　　注

＊1：定时器、累加定时器、计数器的触点・线圈是位软元件，当前值为字软元件。
＊2：实际可以使用的点数根据智能功能模块的不同而所有不同。
　　缓冲存储器点数的有关内容请参照所使用的智能功能模块的手册。
＊3：输入、输出、步进继电器、通信用特殊继电器、通信用特殊寄存器保持默认值原封不动，不可进行更改。

表10.1软元件一览表

<table>
<tr><th rowspan="2">分类</th><th rowspan="2">类别</th><th rowspan="2" colspan="2">软元件名称</th><th colspan="2">默认值</th><th rowspan="2">采用参数设置的设置范围</th><th rowspan="2">参照</th></tr>
<tr><th>点数</th><th>使用范围</th></tr>
<tr><td rowspan="6">直接通信软元件</td><td rowspan="4">位软元件</td><td colspan="2">通信输入</td><td>8192点</td><td>Jn\X0～Jn\X1FFF</td><td rowspan="6">不可</td><td rowspan="6">10.4节</td></tr>
<tr><td colspan="2">通信输出</td><td>8192点</td><td>Jn\Y0～Jn\Y1FFF</td></tr>
<tr><td colspan="2">通信继电器</td><td>16384点</td><td>Jn¥B0～Jn¥B3FFF</td></tr>
<tr><td colspan="2">通信特殊继电器</td><td>512点</td><td>Jn¥SB0～Jn¥SB1FF</td></tr>
<tr><td rowspan="2">字软元件</td><td colspan="2">通信寄存器</td><td>16384点</td><td>Jn¥W0～Jn¥W3FFF</td></tr>
<tr><td colspan="2">通信特殊寄存器</td><td>512点</td><td>Jn¥SW0～Jn¥SW1FF</td></tr>
<tr><td>智能功能模块软元件</td><td>字软元件</td><td colspan="2">缓冲寄存器</td><td>65536点</td><td>Un\G0～Un\G65535*2</td><td>不可</td><td>10.5节</td></tr>
<tr><td>变址寄存器</td><td>字软元件</td><td colspan="2">变址寄存器</td><td>10点</td><td>Z0～Z9</td><td>不可</td><td>10.6节</td></tr>
<tr><td rowspan="2">文件寄存器</td><td rowspan="2">字软元件</td><td rowspan="2">文件寄存器</td><td>Q00JCPU</td><td>0点</td><td>—</td><td rowspan="2">不可</td><td rowspan="2">10.7节</td></tr>
<tr><td>Q00CPU/Q01CPU</td><td>32k点</td><td>R0～R32767
ZR0～ZR32767</td></tr>
<tr><td>嵌套</td><td>—</td><td colspan="2">嵌套</td><td>15点</td><td>N0～N14</td><td>不可</td><td>10.8节</td></tr>
<tr><td rowspan="2">指针</td><td rowspan="2">—</td><td colspan="2">指针</td><td>300点</td><td>P0～P299</td><td rowspan="2">不可</td><td>10.9节</td></tr>
<tr><td colspan="2">中断指针</td><td>128点</td><td>I0～I127</td><td>10.10节</td></tr>
<tr><td rowspan="3">其他</td><td rowspan="3">—</td><td colspan="2">网络号</td><td>239点</td><td>J1～J239</td><td rowspan="3">不可</td><td>10.11.1项</td></tr>
<tr><td rowspan="2">输入/输出号</td><td>Q00JCPU</td><td>—</td><td>U0～UF</td><td rowspan="2">10.11.2项</td></tr>
<tr><td>Q00CPU/Q01CPU</td><td>—</td><td>U0～3F</td></tr>
<tr><td rowspan="3">常数</td><td rowspan="3">—</td><td colspan="2">十进制常数</td><td colspan="3">K-2147483648～K2147483647</td><td>10.12.1项</td></tr>
<tr><td colspan="2">十六进制常数</td><td colspan="3">H0～HFFFFFFFF</td><td>10.12.2项</td></tr>
<tr><td colspan="2">字符串常数</td><td colspan="3">“ABC”，“123” *5</td><td>10.12.3项</td></tr>
</table>

要　　点

*5：字符串只有$MOV指令可以使用。
　　其他指令不可使用字符串。

备　　注

*2：实际可以使用的点数根据智能功能模块的不同而所有不同。
　　缓冲存储器点数的有关内容请参照所使用的智能功能模块的手册。

10.2 内部用户软元件

内部用户软元件就是可以根据用户的用途使用的软元件。

内部用户软元件已预先设置可以使用的点数（默认值）。
通过PC参数的元件设置，可以更改使用点数。

【元件设置画面】

Qn(H)パラメータ設定

プログラム設定 | ブートファイル設定 | SFC設定 | I/O割付設定 | シリアルコミュニケーション設定
PCネーム設定 | PCシステム設定 | PCファイル設定 | PC RAS設定 | デバイス設定

	記号	進	デバイス点数	ラッチ(1)先頭	ラッチ(1)最終	ラッチ(2)先頭	ラッチ(2)最終	ローカルデバイス先頭	ローカルデバイス最終
入力リレー	X	16	2K						
出力リレー	Y	16	2K						
内部リレー	M	10	8K						
ラッチリレー	L	10	2K						
リンクリレー	B	16	2K						
アナンシェータ	F	10	1K						
リンク特殊	SB	16	1K						
エッジリレー	V	10	1K						
ステップリレー	S	10	2K						
タイマ	T	10	512						
積算タイマ	ST	10	0K						
カウンタ	C	10	512						
データレジスタ	D	10	11136						
リンクレジスタ	W	16	2K						
リンク特殊	SW	16	1K						

デバイス合計 16.4 Kワード
ワードデバイス 15.2 Kワード
ビットデバイス 19.0 Kビット

デバイス点数の合計は16kワードまでです。
ラッチ(1):ラッチクリアにてクリアが可能です。
ラッチ(2):ラッチクリアにてクリアが不可です。
PCメモリクリアにてクリアを行ってください。

X/Y割付確認 | マルチCPU設定 | デフォルト | チェック | 設定終了 | キャンセル

默认值
[ ]内表示点数的元件
的使用点数可以更改。

（1）内部用户软元件的设置范围

基本型QCPU除输入（X）、输出（Y）、步进继电器（S）、通信特殊继电器（SB）、通信特殊寄存器（SW）以外的内部用户软元件可以在PC参数的软元件设置时在16.4 k字（包括上述软元件所占的1.5 k字）的范围内更改使用点数。
更改内部用户软元件点数时的思路说明如下。

（a）关于设置范围

① 每1软元件采用16点单位进行设置。

② 每1软元件的最大点数位32 k。
定时器、保持定时器、计数器的1点算作线圈1点加上触点1点，合计2点。

（2）存储容量的思路

内部用户软元件的设置应满足以下的式子。

1.5＋（位软元件容量）＋（字软元件容量）＋（定时器、保持定时器、计数器容量）≤16.4 k

(a) 位软元件时

位软元件将16点作为1个字计算。

$$\text{（位软元件容量）} = \frac{\text{（M+L+F+V+B 的合计数）}}{16} \quad \text{（字）}$$

(b) 定时器（T）、保持定时器（ST）、计数器（C）时

定时器（T）、保持定时器（ST）、计数器（C）将16点作为18个字计算。

定时器、保持定时器、计数器容量＝（T　ST　C的合计数）/ 16　18（字）

(c) 字软元件时

数据寄存器（D）、通信寄存器（W）将16点作为16个字计算。

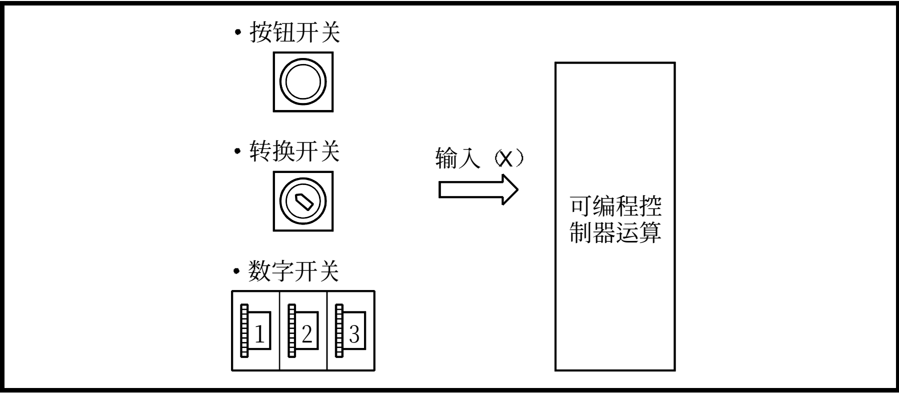
字软元件容量=（D　W的合计数）/ 16　16（字）

要　　点
（1）利用参数更改内部用户软元件的使用点数时，不可原封不动地使用采用更改前的参数编写的可编程控制器程序。 更改内部用户软元件的使用点数后，请从基本型QCPU读出顺控程序存入GX Developer，然后再重新写入基本型QCPU。

10.2.1 输入（X）

（1）什么是输入

(a) 输入就是通过按钮、转换开关、限位开关、数字开关等的外部器件给予基本型QCPU的指令或数据。



(b) 将每1点输入假设成基本型QCPU内置的继电器Xn，在程序中使用该Xn的a触点、b触点。

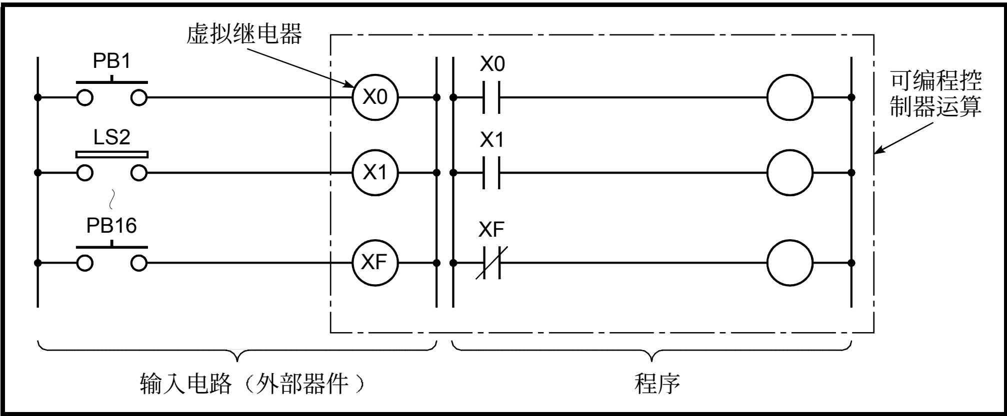


图10.1 输入（X）的分析

(c) 程序内Xn的a触点和b触点的使用数只要在程序容量的范围内，就没有限制。

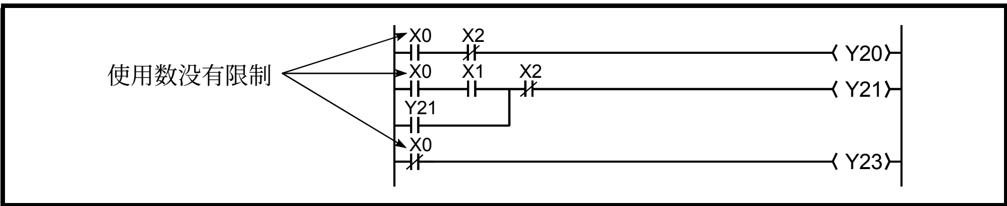
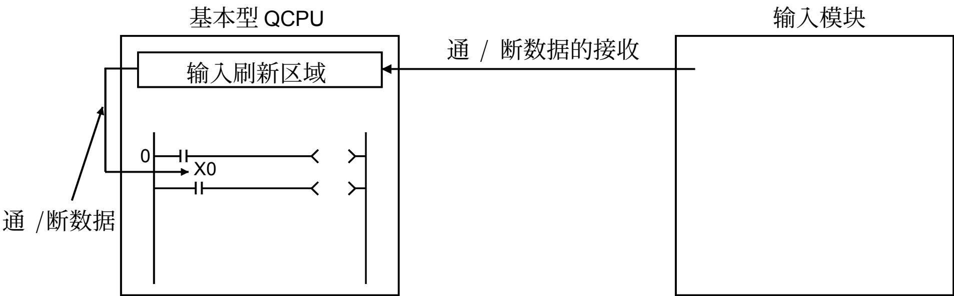


图10.2 输入（X）在程序上的使用

（2）输入的接收方法

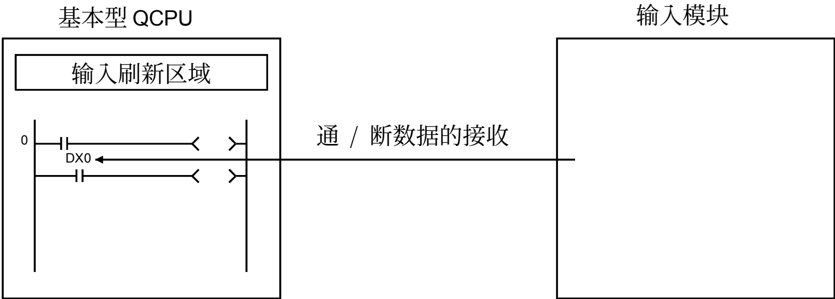
（a）输入有“刷新输入”和“直接存取输入”2种。

① 刷新输入就是采用顺控程序执行前的输入刷新时接收的通/断数据进行运算的输入方式。＊1



顺控程序中用X□指定。
例如，输入的10就写成X10。

② 直接存取输入就是采用指令执行时从输入模块接收的通/断数据进行运算的输入方式。



顺控程序中用X□指定。
例如，输入的10就写成D10。
直接存取输入可以使用以1点为单位使用输入的指令（LD、AND、OR等）。

（b）刷新输入和直接存取输入的不同点

直接存取输入在指令执行时从直接输入器件上进行存取，因此，和刷新输入相比，输入的接收较快。
但是，和刷新输入相比，指令的处理时间将加长。
而且，直接存取输入只可以用在主基板及扩展基板上安装的输入模块、智能功能模块所使用的输入上。
刷新输入和直接存取输入的不同点如表10.2所示。

表10.2 刷新输入和直接存取输入的不同点一览表

项　　目	刷新输入	直接存取输入
主/扩展基板上安装的输入模块	可以使用	可以使用
主/扩展基板上安装的智能功能模块的输入		
MELSECNET/H网络系统、CC—Link系统所使用的输入	可以使用	不可使用

备　　注

＊1：刷新方式的详细内容请参照4.7.1项。

（c）顺控程序中不可将同一输入编号指定为刷新输入和直接存取输入。
使用直接存取输入后再使用刷新输入的情况下，采用直接存取输入所接收的通/断数据进行运算。
采用结束处理的输入刷新时所

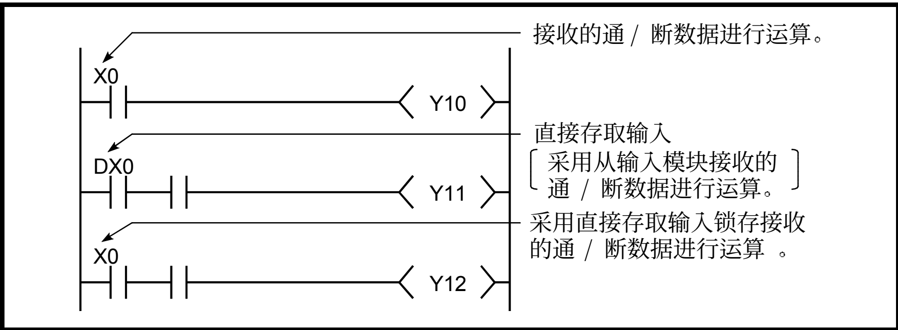


图10.3 刷新输入和直接存取输入

要　　点

（1）对编写的程序进行调试时，可以采用以下方法通/断输入（X）。

- OUT Xn指令

ON/OFF指令　　X1　　OUT X1

- GX Developer的测试操作

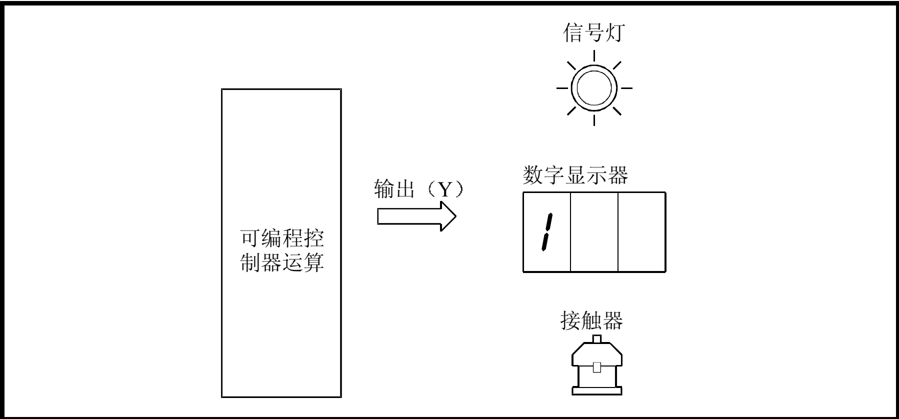
（2）输入（X）也可以用于以下场合

- CC—Link的Rx的刷新对象（基本型QCPU侧）的软元件
- MELSECNET/H的通信输入的刷新对象（基本型QCPU侧）的软元件

10.2.2 输出（Y）

（1）什么是输出

(a) 输出就是将控制结果输出到外部的信号灯、数字显示器、电磁开合器(接触器)、螺线管等。



（b）输出可以以相当于1a触点的形式向外部取出。

（c）程序中的输出Yn的a触点和b触点的使用数只要在程序容量的范围内，就没有限制。

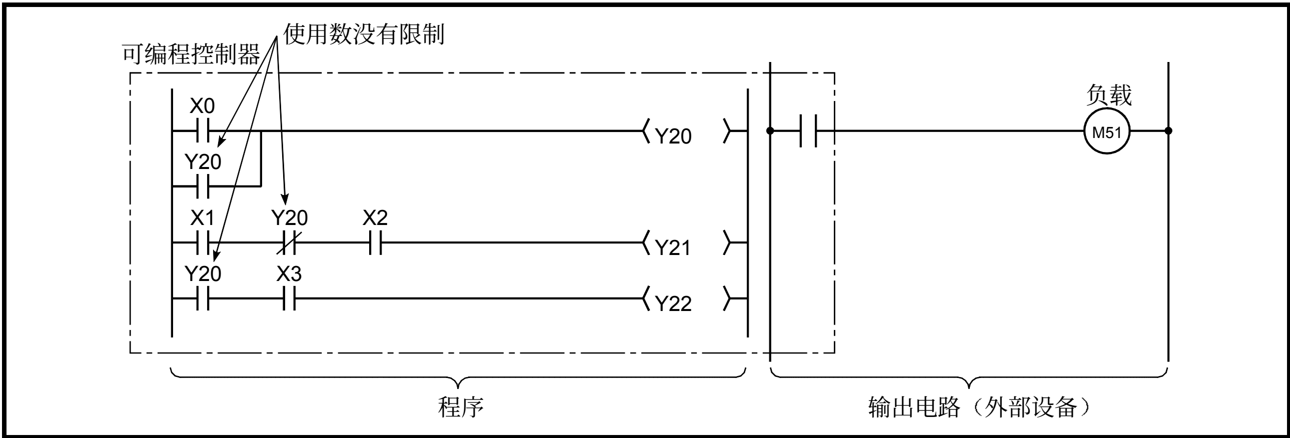
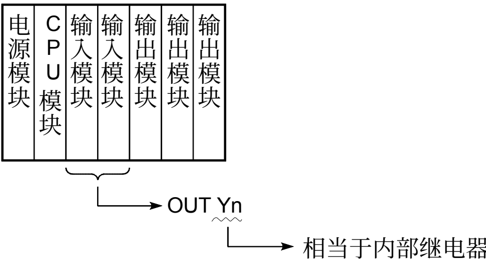


图10.4 输出（Y）的分析

（2）作为内部继电器（M）的代用品使用

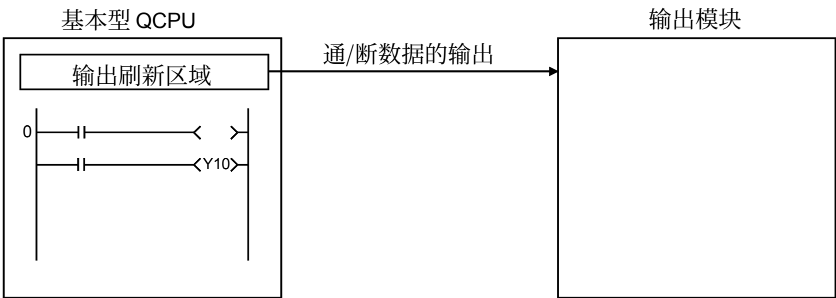
与安装输入模块的领域以及未安装模块领域相对应的“Y”可以作为内部继电器M的代用品使用。



（3）将输出向外部输出的方法

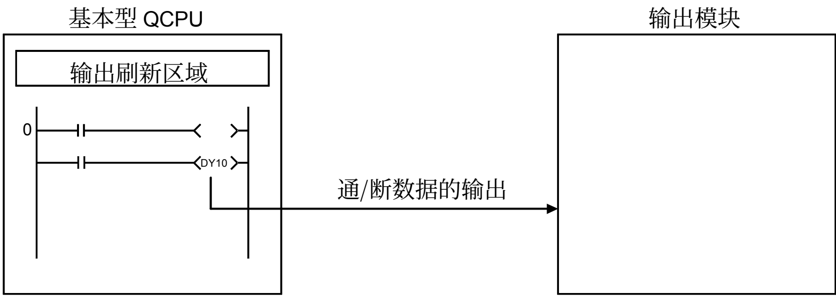
（a）输出有2种方法，即“刷新输出”和“直接存取输出”。

① 刷新输出就是在顺控程序执行前的输出刷新时将通/断数据输出到输出模块。＊1



顺控程序中用Y□指定。
例如，输出的10写作Y10。

② 直接存取输出就是在指令执行时将通/断数据输出到输出模块上。



顺控程序中用DY□指定。
例如，输出的10写作DY10。

（b）刷新输出和直接存取输出的不同点

直接存取输出是在指令执行时直接和输出模块进行存取，因此，开始向外部输出之前的时间将缩短。
但是，和刷新输出相比，指令的处理时间将会延长。
而且，直接存取输出只可以用在基本型上安装的输出模块、智能功能模块所使用的输出上。
刷新输出和直接存取输出的不同点如表10.3所示。

表10.3 刷新输出和直接存取输出的不同点一览表

项　　目	刷新输入	直接存取输入
主/扩展基板上安装的输出模块 主/扩展基板上安装的智能功能模块的输出	可以使用	可以使用
MELSECNET/H网络系统、CC—Link系统所使用的输入	可以使用	不可使用

备　　注

＊1：刷新方式的详细内容请参照4.7.1项。

10.2.3 内部继电器（M）

（1）内部继电器

(a) 内部继电器就是基本型QCPU内部所使用的不可锁存（停电保持）的辅助继电器。

进行以下操作的内部继电器全部断。

- 从电源断的状态打开电源时
- 基本型QCPU复位时
- 基本型QCPU锁存清零时

(b) 程序中的触点（a触点、b触点）的使用数只要在程序容量的范围内，就没有限制。

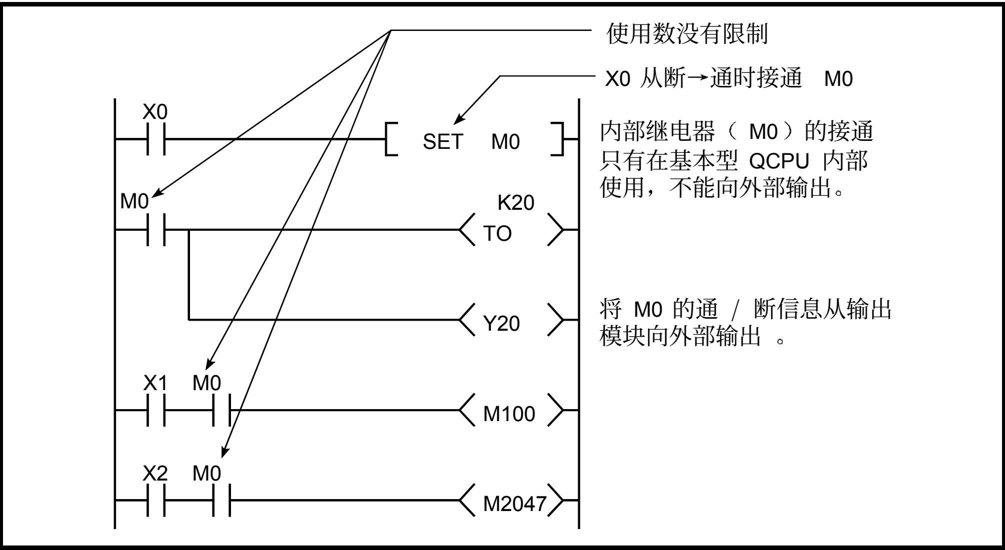


图10.5 内部继电器

（2）向外部的输出方法

将顺控程序的运算结果向外部输出时，使用输出（Y）。

备　注

1）需要锁存（停电保持）的情况下，请使用锁存继电器（L）。
锁存继电器的有关内容请参照10.2.4项。

10.2.4 锁存继电器（L）

（1）什么是锁存继电器

（a）锁存继电器就是基本型QCPU内部所使用的能够进行锁存（停电保持）的辅助继电器。

锁存继电器即使进行以下操作，运算结果（通/断信息）也会保持。

- 从电源切断的状态打开电源时
- 基本型QCPU进行复位操作时

锁存利用基本QCPU主体的电池进行。

（b）如果GX Developer上进行远程锁存清零，就可以断锁存继电器。

但是，在PC参数的软元件设置时设置为“锁存（2）：不可利用锁存清零进行清零”的锁存继电器即使进行远程锁存清零，也不能切断。

（c）程序中的触点（a触点、b触点）的使用数只要在程序容量的范围内，就没有限制。

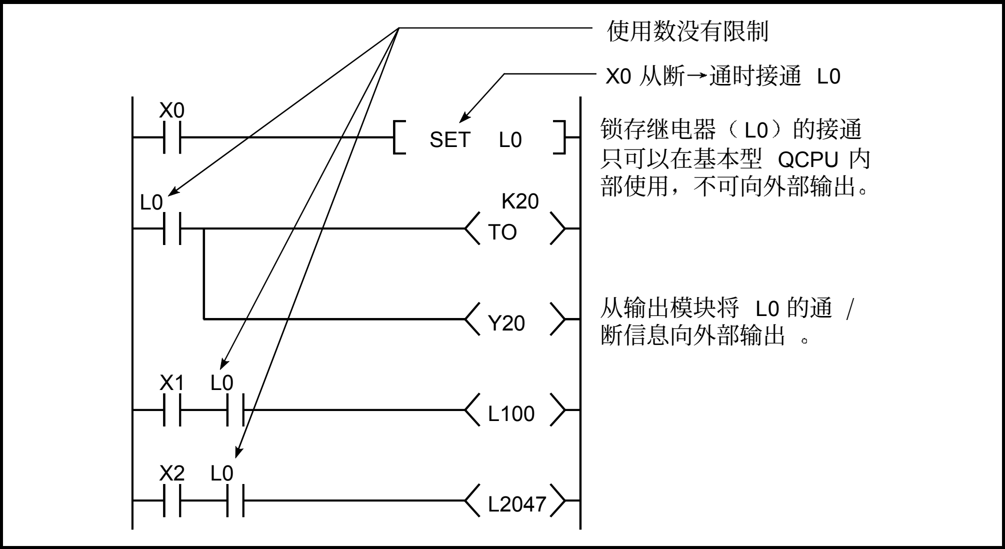


图10.6 锁存继电器

（2）向外部输出的方法

将顺控程序的运算结果向外部输出时，使用输出（Y）。

备　注

需要锁存的情况下，请使用内部继电器（M）。

内部继电器的有关内容请参照10.2.3项。

10.2.5 信号报警器（F）

（1）什么是信号报警器

（a）信号警报器就是用在用户编写的设备异常、故障检测程序中十分方便的内部继电器。

（b）信号警报器接通时，特殊继电器（SM62）接通，特殊寄存器（SD62～SD79）中存入信号报警器的个数和编号。

此时，“ERR.”LED点亮。

• 特殊继电器　：SM62………………只要有1个信号报警器接通即接通。

• 特殊寄存器　：SD62………………存入最早接通的信号报警器编号。

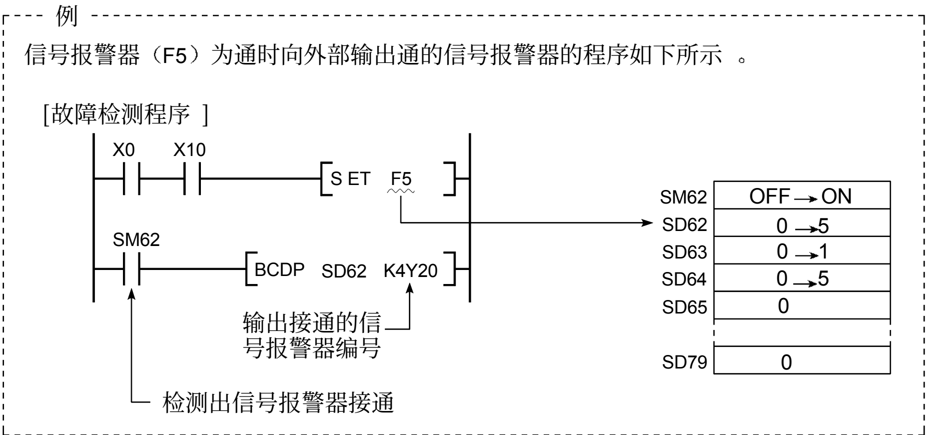
SD63……………… 存入接通的信号报警器个数。

SD64～SD79………按照接通的先后次序存入信号报警器的编号，（SD62和SD64存入同一信号报警器的编号。）

此外，SD62中存储的信号报警器编号也注册在故障记录存储区域内。

但是，可编程控制器接通期间，故障记录存储区域内存储的信号报警器编号只有1个。

（c）如果故障检测程序中使用信号报警器，特殊继电器（SD62）接通时只要监视特殊寄存器（SD62～SD79），就可以确认设备的故障和故障的有无（信号报警器编号）。



（2） 接通信号报警器的方法

（a）接通信号报警器的方法

信号报警器可以利用SET F□指令、OUT F□指令进行操作。

① SET F□指令仅在输入条件上升（断→通）时接通信号报警器，此后输入条件即使断信号报警器也保持接通。
大量使用信号报警器的情况下，通过使用OUT F□指令，可以加快扫描时间。

② 使用OUT F□指令可以通/断信号报警器，但因为每次扫描都要进行处理，所以比使用SET F□指令要来得慢。
此外，即使利用OUT F□指令切断信号报警器，因为还必须执行RST F□指令/BKRST指令，所以，请使用SET F□指令接通信号报警器。

要　　点
（1）使用SET F□指令或OUT F□指令以外的指令（例如MOV指令）接通信号报警器时，和内部继电器的动作相同。 （SM62接通以及SD62不向SD64～SD79存入信号报警器的编号。）

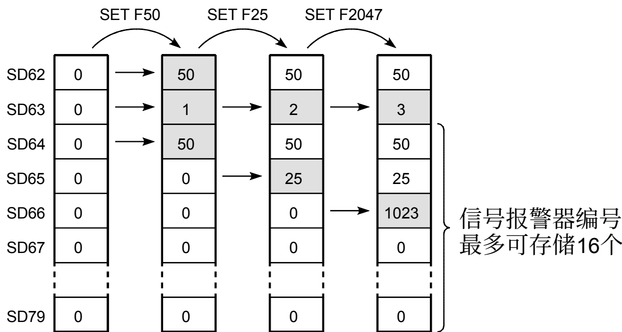
（b）接通信号报警器时的处理

① 特殊寄存器（SD62～SD79）的存储数据

a）接通的信号报警器编号依次存入SD64～SD79。

b）SD64中存储的信号报警器编号存入SD62。

c）SD63的内容＋1。



② 主体上的LED显示
信号报警器接通时，基本型QCPU正面的“ERR.”LED点亮。

（3） 切断信号报警器的方法和处理内容

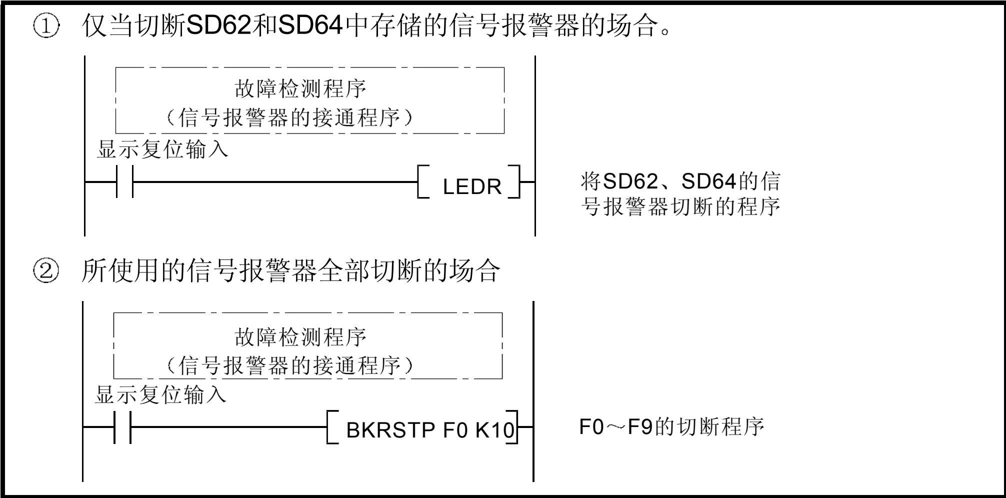
（a）切断信号报警器的方法

切断信号报警器可以利用RST F□指令、BKRST指令以及OUT F□指令进行。

① RST F□指令用于将利用SET F□指令接通的信号报警器编号切断的场合。

② BKRST指令用于将指定范围内的信号报警器编号一次性地切断的场合。

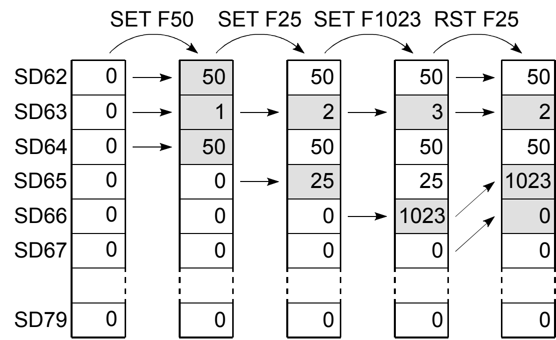
③ OUT F□指令可以用同一指令通/断信号报警器。
但是，即使用OUT F□指令将信号报警器编号切断，也不进行（b）的信号报警器切断时的处理。
利用OUT F□指令将信号报警器编号切断后，必须执行RST F□指令或BKRST指令。



备　注

RST指令、BKRST指令的详细内容请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

（b）信号报警器切断时的处理内容
① 执行RST F□指令将信号报警器切断时的特殊寄存器（SD62～SD79））的存储数据
• 切断的信号报警器的编号被删除，存储在被删除的信号报警器以后的信号报警器编号向前移。
• SD64中存储的信号报警器编号切断时，新存入SD64中的信号报警器编号也存入SD62中。
• SD63的内容－1。
• SD63为0时，切断SM62。



	初始	SET F50	SET F25	SET F1023	RST F25
SD62	0	50	50	50	50
SD63	0	1	2	3	2
SD64	0	50	50	50	50
SD65	0	0	25	25	1023
SD66	0	0	0	1023	0
SD67	0	0	0	0	0
SD79	0	0	0	0	0

② 执行BKRST指令将切断信号报警器时的特殊寄存器（SD62～SD79）的存储数据

• 利用BKRST指令指定的信号报警器编号被删除，存储在被删除信号报警器以后的信号报警器编号向前移。
• SD64中存储的信号报警器编号断时，新存入SD64中的信号报警器编号也存入SD62。
• SD63的内容减去被复位的信号报警器数。
• SD63为0时，切断SM62。

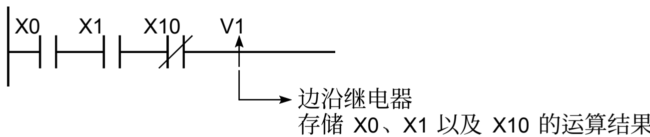
③ LED显示

SD64～SD79的信号报警器编号全部切断时，“.ERR.”LED熄灭。

10.2.6 边沿继电器（V）

（1）什么是边沿继电器

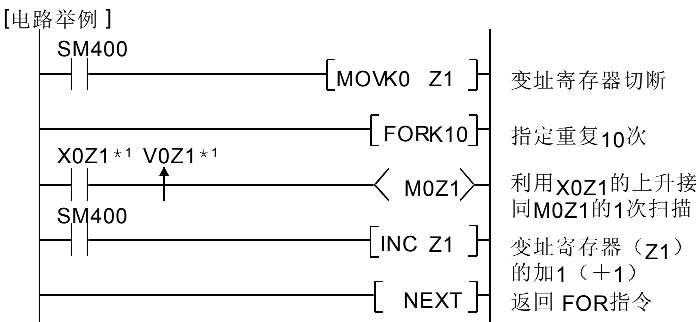
(a) 边沿继电器就是存储从电路块头部开始的运算结果（通/断信息）的软元件，只可以在触点上使用。

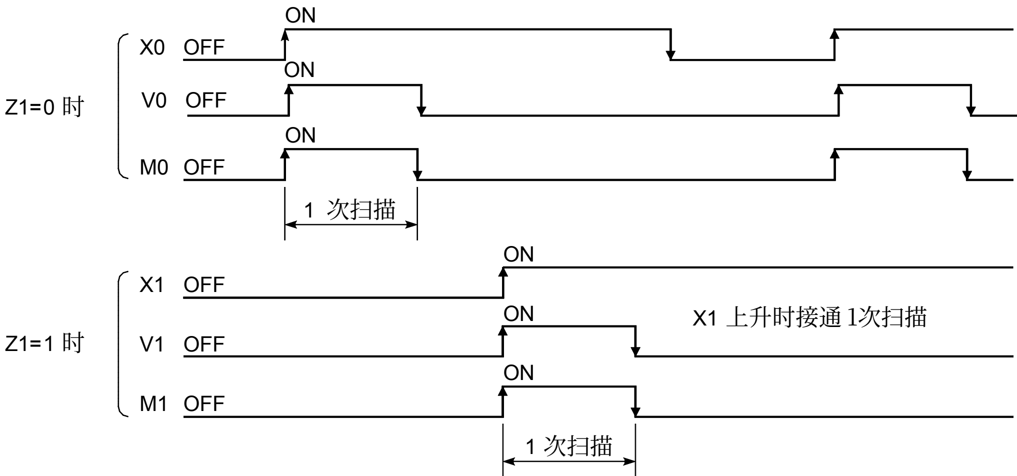


（b）基本型QCPU上执行的程序中，不可使用同一边沿继电器编号。

（2）边沿继电器的用途

边沿继电器可用于变址修饰的结构化程序采用检出上升（断→通）时执行的场合。





备　注

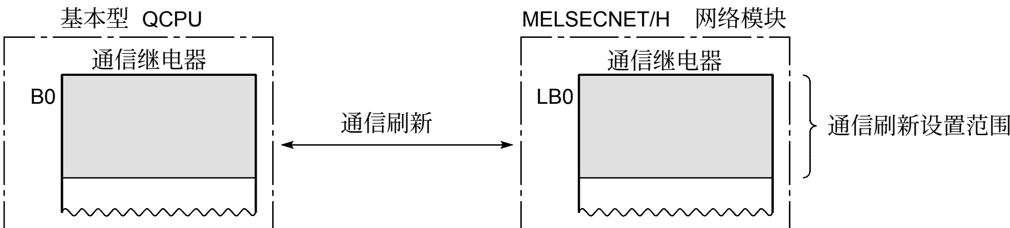
＊1：X0Z1的通/断信息存入边沿继电器的V0Z1。

例如，将X0的通/断信息存入V0，将X0的通/断信息存入V1。

10.2.7 通信继电器（B）

（1）什么是通信继电器

(a) 通信继电器就是MELSECNET/H网络模块等的通信继电器（LB）对基本型QCPU进行刷新时，或者基本型QCPU内的数据对MELSECNET/H网络模块等的通信继电器（LB）进行刷新时基本型QCPU一侧的继电器。



MELSECNET/H网络系统等上不使用的范围不可作为内部继电器/锁存继电器使用。

- 通信继电器上不进行锁存的范围・・相当于内部继电器
- 通信继电器上进行锁存的范围・・・・相当于锁存继电器

(b) 程序内触点（a触点、b触点）的使用数只要在程序容量的范围内就没有限制。

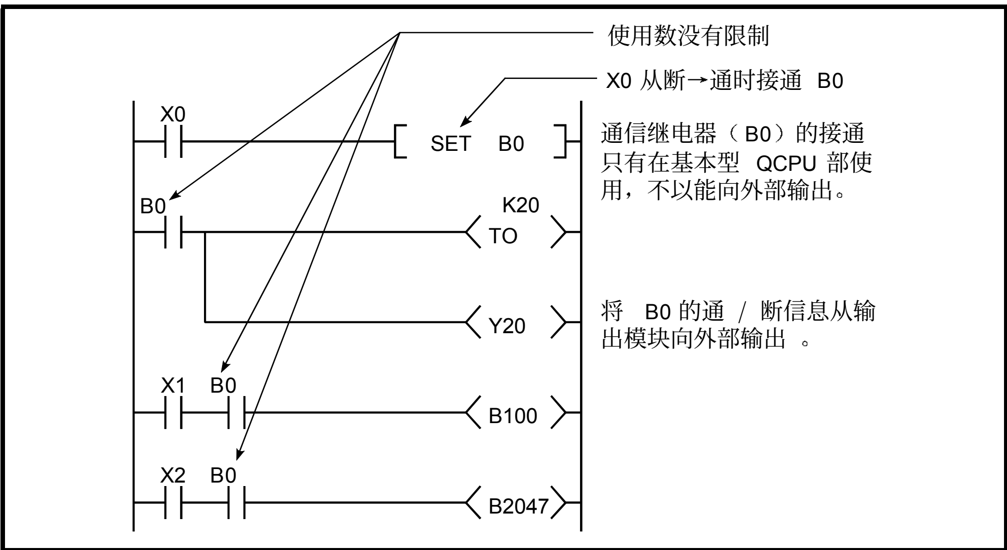


图10.5 通信继电器

（2）网络上使用时

网络上使用时，必须进行网络参数的设置。

备　注

1）网络参数的有关内容请参照下列手册。
- Q对应MELSECNET/H网络系统参考手册

2）MELSECNET/H网络模块内的通信继电器为16384点，而基本型QCPU的通信继电器为2048点。基本型QCPU上使用2048点以后的通信继电器时，请在PC参数的软元件设置时更改通信继电器的点数。

10.2.8 通信用特殊继电器（SB）

（1）什么是通信特殊继电器

（a）通信特殊继电器就是表示MELSECNET/H网络模块等的智能功能模块的通信状态和异常检出情况的继电器。

（b）根据数据通信时所发生的各种各样的原因控制通/断，并通过对通信特殊继电器的监视，可以掌握数据通信的通信状态和异常状态等。

（2）通信特殊继电器的点数

通信特殊继电器为SB0～SB3FF的1024点，MELSECNET/H网络模块等的每个智能功能模块有512点。

备　注

可以使用的通信特殊继电器的详细内容请参照下列手册。

- QCPU（Q模块）/QnACPU编程手册（通用指令篇）

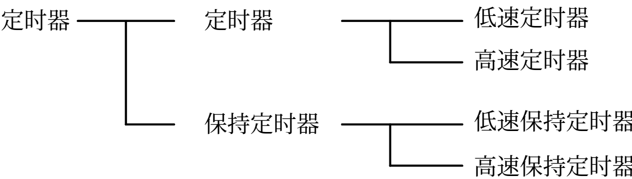
10.2.9 步进继电器（S）

步进继电器就是SFC程序所用的软元件。
步进继电器不可用于将来的扩展。

10.2.10 定时器（T）

定时器是加法式的，在定时器线圈接通时开始计测，当前值超出设置值定时器到达时间时，触点接通。
定时器到达时间时，当前值和设置值为同一值。

定时器有线圈切断时当前值为0的定时器和即使线圈切断仍保持当前值的保持定时器2种类型。
定时器分低速定时器和高速定时器，保持定时器分低速保持定时器和高速保持定时器。

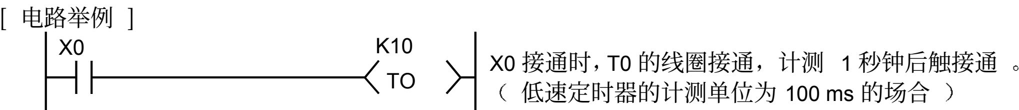


低速定时器和高速定时器是相同的软元件，由定时器指定（指令的写入）而成为低速定时器或高速定时器。例如，指定OUT T0时为低速定时器；指定OUTH T0时则为高速定时器。
低速保持定时器和高速保持定时器是相同的软元件，由定时器指定（指令的写入）而成为低速保持定时器或高速保持定时器。例如，指定OUT ST0时为低速保持定时器；指定OUTH ST0时则为高速保持定时器。

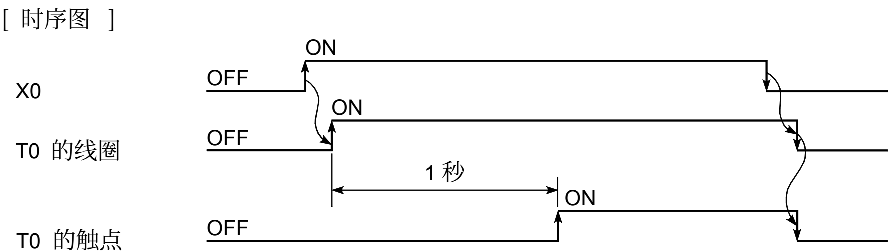
低速定时器

（1）什么是低速定时器
（a）低速定时器是仅当线圈接通期间才有效的定时器。
（b）定时器的线圈接通时开始计测，定时器达到时间时触点接通。
定时器的线圈切断时当前值为0，触点也切断。

[电路举例]



[时序图]



（2）计测单位
（a）低速定时器的计测单位（时限）的默认值为100 ms。
（b）计测单位可以在1～1000 ms的范围内，以1 ms单位变化。
其设置在PC参数的PC系统设置时进行。

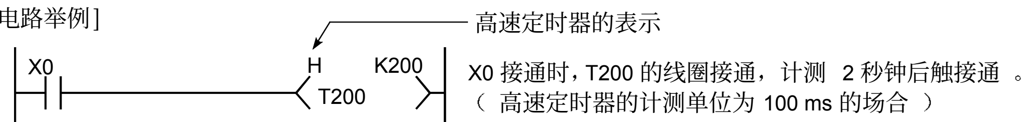
高速定时器

（1）什么是高速定时器

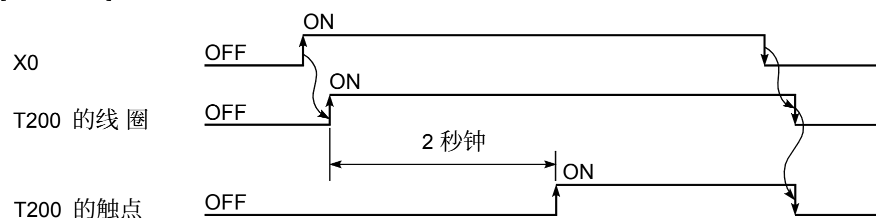
（a）高速定时器是仅在线圈接通期间有效的定时器，附加“H”记号。

（b）定时器的线圈接通时开始计测，达到时间时触点接通。
定时器的线圈切断时当前值为0，触点也切断。

[电路举例]



[时序图]



（2）计测单位

（a）高速定时器的计测单位（时限）的默认值为10 ms。

（b）计测单位可以在0.1～100 ms的范围内，以0.1 ms单位变化
其设置在PC参数的PC设置时进行。

保持定时器

（1）什么是保持定时器

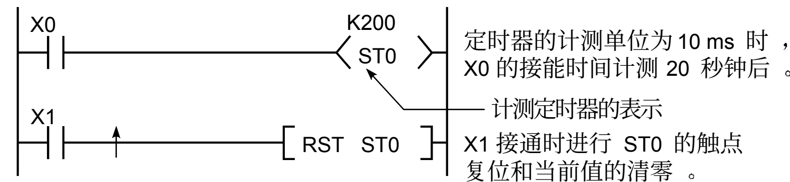
（a）保持定时器就是计测线圈接通时间的定时器。

（b）定时器的线圈接通时开始计测，达到时间时触点通。
定时器的线圈切断时保持当前值和触点的通/断状态。
线圈再次接通时，从保持的当前值重新开始计测。

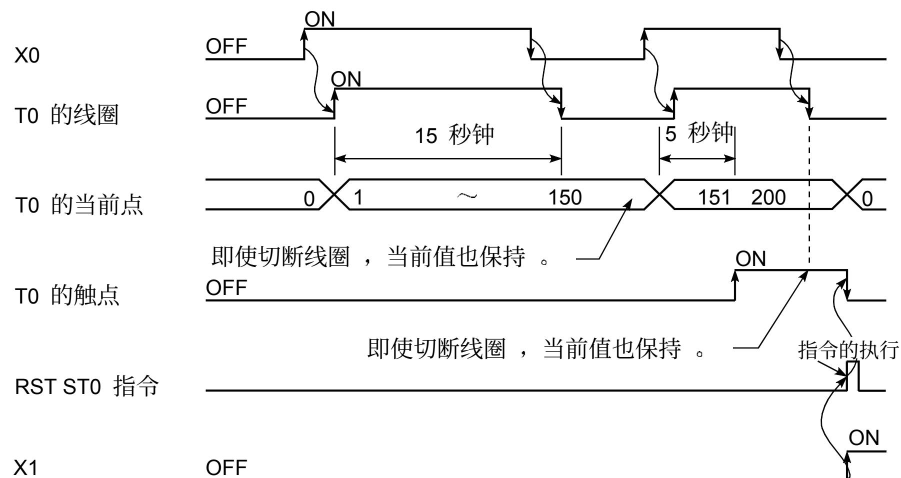
（c）保持定时器有低速保持定时器和高速保持定时器2种类型。

（d）当前值的清零和触点的切断利用RST T□指令进行。

[电路侧]



[时序图]



（2）计测单位

（a）保持定时器的计测单位（时限）和低速定时器、高速定时器相同。
- 低速保持定时器：低速定时器
- 高速保持定时器：高速定时器

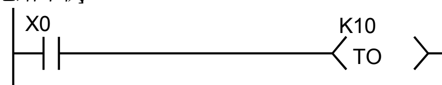
备 注

使用保持定时器时，必须通过PC参数的软元件设置对保持定时器的使用点数进行设置。

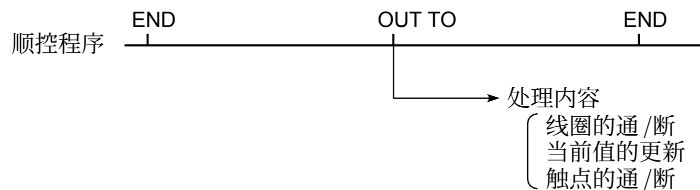
定时器的处理方法和精度

(a) OUT T□指令执行时进行定时器线圈的通/断、当前值的更新以及触点的通/断处理。
结束处理时不进行定时器当前值的更新和触点的通/断处理。

[电路举例

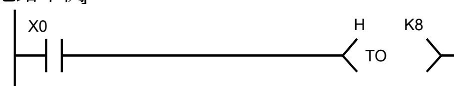


[OUT T0 指令执行时的处理内容]

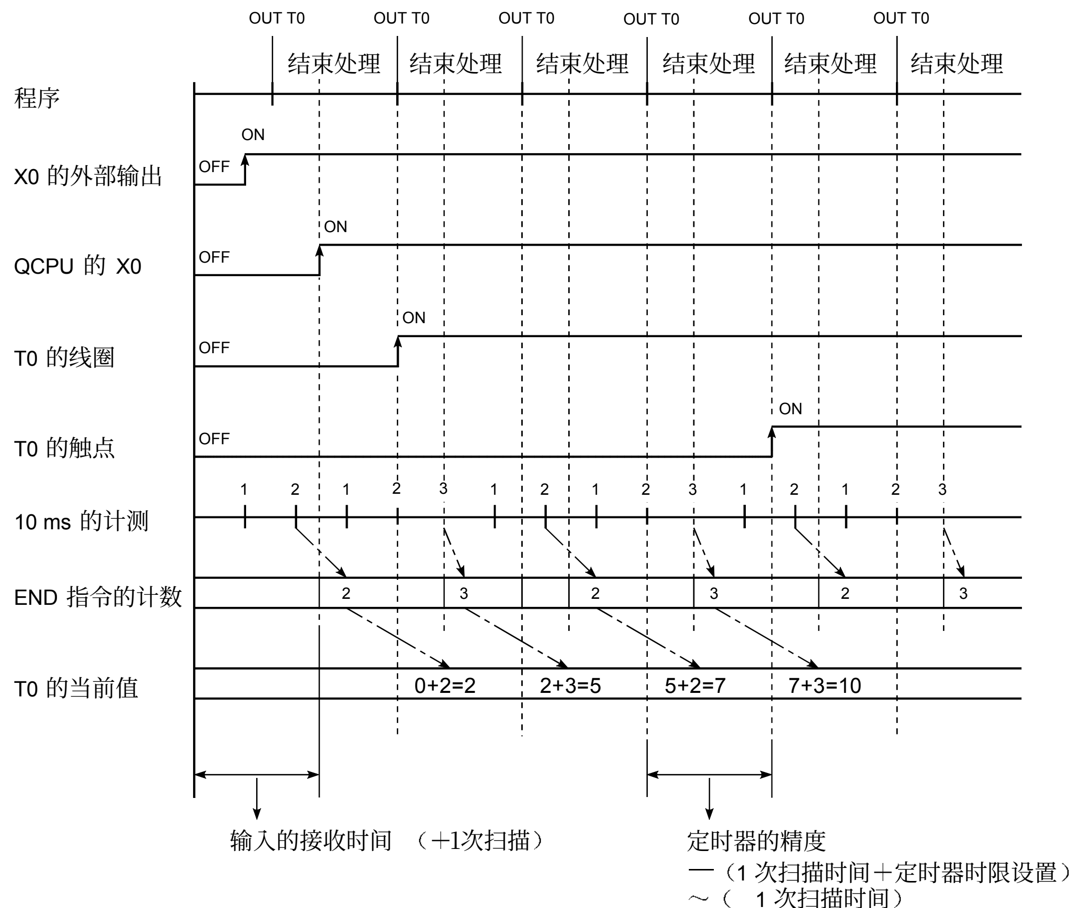


(b) 当前值在OUT T□指令执行时要加上END指令时计测所得的扫描时间。
OUT T□指令执行时定时器的线圈为断的情况下，不进行当前值的更新。

[电路举例]



[当前值的更新时序]

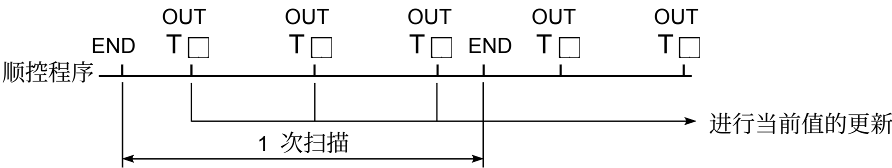


（c）接收输入（X）到输出为止的定时器的响应精度为：最大值＋（2次扫描＋定时器的时限设置）。

定时器使用时的注意事项

定时器使用时的注意事项如下所示。

(a) 1次扫描期间，同一定时器不可用多个OUT T□指令记述。
同一定时器采用多个OUT T□指令记述时，各个OUT T□指令执行时定时器的当前值将被更新，这样，就无法进行正常的计测。



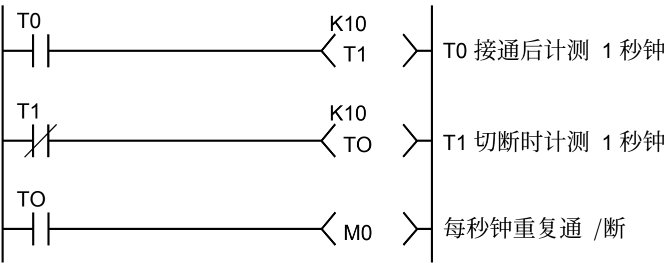
（b）定时器（例：T1）D的线圈接通期间，不可用CJ指令等跳过OUT T1指令。
跳过OUT T□指令时，定时器的当前值将得不到更新。

（c）定时器不可在中断程序中使用。

（d）定时器得设置值为0的情况下，OUT T□指令执行时触点接通。

（e）定时器达到时间后，即使将设置值更改为比当前值更大得值，定时器仍保持时间达到时的状态，不再动作。

（f）采用2个定时器建立通/断电路时，请采用如下图所示的电路。



10.2.11 计数器（C）

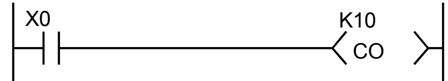
计数器是加法式的，当计数值和设置值相同时计数到达，触点接通。
计数器有2种类型，一是对顺控程序输入条件的上升次数进行计数的计数器；二是对中断原因的发生次数进行计数的中断计数器。

计数器

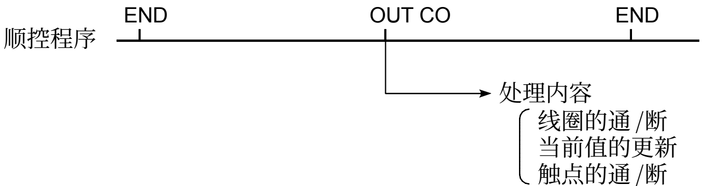
（1）什么是计数器
计数器就是对顺控程序输入条件的上升次数进行计数的软元件。

（2）计数处理
(a) OUT C□指令执行时进行计数器线圈的通/断、当前值的更新（计数值＋1）以及触点的通/断处理。
结束处理期间不进行计数器当前值的更新和触点的通/断处理。

[电路举例

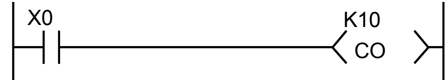


[OUT C0 指令执行时（X0：断→通时） 的处理内容]

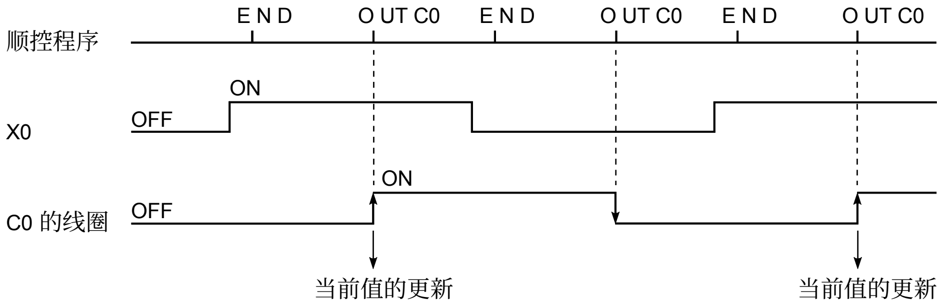


(b) 当前值的更新（计数值＋1）在OUT C□指令的上升时（断→通）进行。
OUT C□指令在断、通→通、以及通→断时不进行当前值的更新。

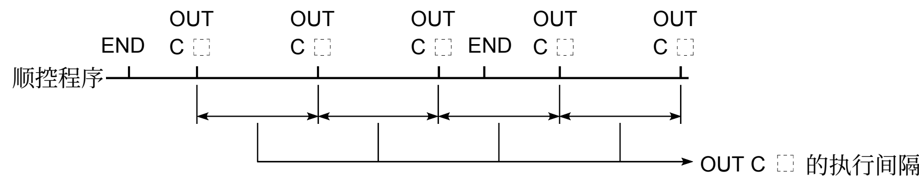
[电路举例



[当前值的更新时序]



（c）1次扫描期间可以记述多个计数器，并提高最大计数速度。
此时计数器的输入信号采用直接存取输入（DX□）。*1

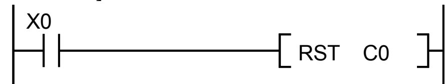


（3）计数器的复位

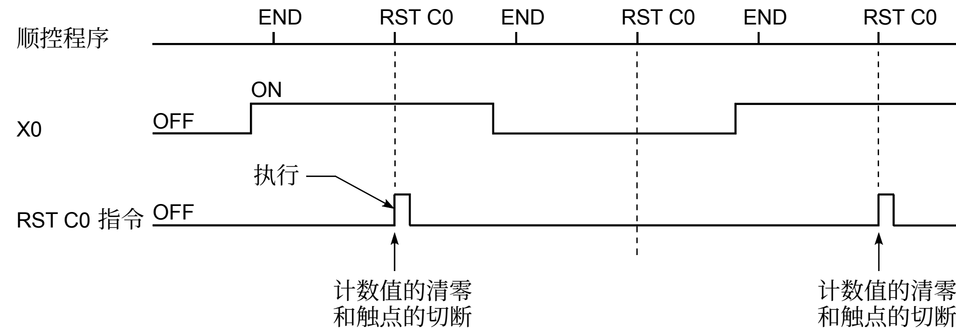
（a）即使OUT C□指令切断，计数器的当前值也不会被清零。
计数器的当前值的清零（复位）和触点的切断采用RST C□指令进行。

（b）执行RST C□指令之时，计数值被清零，触点也被切断。

[电路举例]



[计数器复位的时序]



（4）计数器的最大计数速度
计数器只有在输入条件的通/断时间比同一OUT C□指令的执行间隔更大的情况下才能计数。
计数器的最大计数速度可用下式算出。

$$\text{最大计数速度Cmax} = \frac{n}{100} \times \frac{1}{T} [\text{次/S}]$$

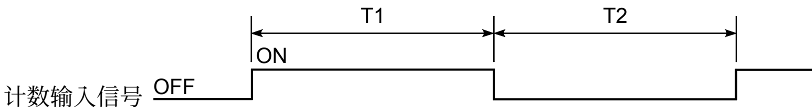
n：效率（%）*2
T：OUT C□指令的执行间隔

备　　注

＊1：直接存取输入的有关内容请参照10.2.1项
＊2：效率（n）是将计数输入信号的通、断时间之比以百分比（%）形式表达所得到的结果。

• T1≥T2 时 $n = \frac{T1}{T1+T2} \times 100\%$

• T1 < T2 时 $n = \frac{T1}{T1+T2} \times 100\%$



中断计数器

（1）什么是中断计数器

中断计数器就是对中断原因的发生次数进行计数的软元件。

（2）中断处理

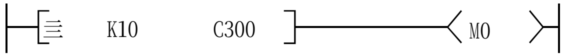
（a）中断计数器在中断发生时对计数器的当前值进行更新。

利用中断计数器进行计数时，不必编写使用中断计数器的程序。

（b）中断计数器即使已指定设置值，也不会计数。

将中断计数器用于控制的情况下，请采用比较指令（＝、＜＝）和设置值进行比较，对内部继电器（M）等进行通/断。

IO中断输入接通10次时接通M0的情况下，编写如下图所示的程序。（对应IO的中断计数器假定为C300。）

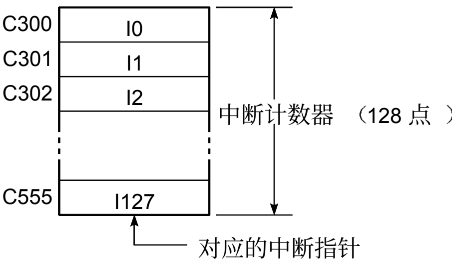


（3）中断计数器的设置

(a) 使用中断计数器的情况下，在PC参数的PC系统设置时设置中断计数器的起始号。

从设置的计数器编号开始的128点成为中断计数器。

中断计数器的起始号设置为C300的情况下，C300～C427成为中断计数器。



（b）使用中断计数器的情况下，在主程序中采用EI指令设为中断允许状态。

（4）注意事项

（a）不可用1个中断指针进行由中断计数器引起的计数和中断程序的执行。
如果在PC参数的PC系统设置时进行中断计数器的设置，就不能执行中断程序。

（b）以下所示的各个处理的执行期间发生中断的情况下，计数处理要等到各个处理的执行结束之后才开始。
各个处理的执行结束之时进行计数处理。
但是，各个处理的过程中即使再次发生同一中断，也只进行1次计数。
• 顺控程序的各个指令的执行期间
• 中断程序执行期间

（c）中断计数器的最大计数速度由以下所示的处理时间中最长的处理所决定。
• 程序中所使用的指令中处理时间最长的指令
• 中断程序的处理时间

（d）如果使用多个中断计数器，顺控程序的处理时间将变长，有时可能出现“WDT ERROR”。
此时，请减少中断计数器的个数，或者降低输入脉冲信号的计数速度。

（e）中断计数器的计数值的复位请利用主程序的RST C□指令进行。

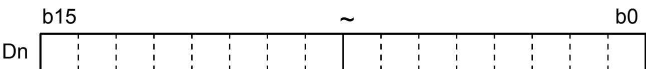
（f）中断计数器的计数值可以利用顺控程序的MOV指令读出。

10.2.12 数据寄存器（D）

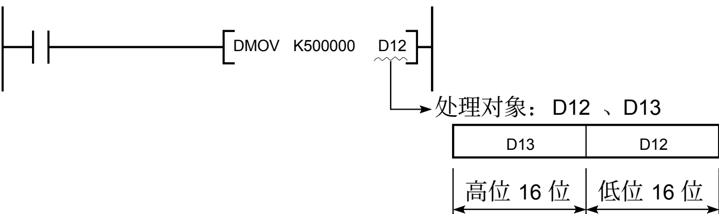
（1 什么是数据寄存器

（a）数据寄存器就是基本型QCPU上可以存储数值数据（—32768～32767，或0000H～FFFFH）的存储器。

（b）每个数据寄存器由16位构成，可以以16位单位读出、写入。



（c）采用32位指令使用数据寄存器时，Dn和Dn＋1将成为处理对象。
顺控程序中指定的数据寄存器编号（Dn）为低位的16位，顺控程序中指定的数据寄存器编号＋1的数据寄存器为高位的16位。例如，采用MOV指令指定D12的情况下，D12为低位的16位；而D13为高位的16位。



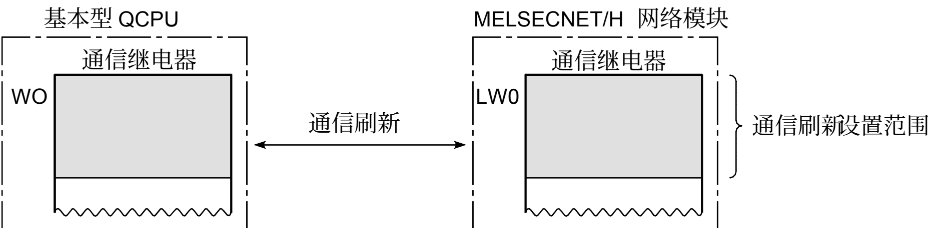
采用2个数据寄存器可以存储—2147483648～2147483647，或0H～FFFFFFFFH的数据。

（d）用顺控程序存储的数据可一直保存到其他数据存入为止。

10.2.13 通信寄存器（W）

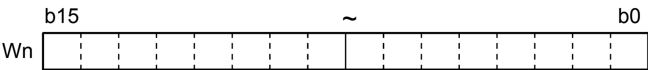
（1）什么是通信寄存器

(a) 通信寄存器就是用MELSECNET/H网络模块等智能功能模块的通信寄存器（LW）的数据刷新基本型QCPU时的基本型QCPU一侧的存储器。
通信寄存器在基本型QCPU上可以存储（—32768～32767，或0000H～FFFFH）的数值数据。

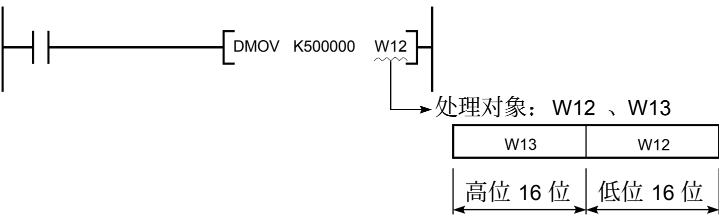


MELSECNET/H网络模块等的不使用范围的通信寄存器可以作为数据寄存器的代用品使用。

（b）每个通信寄存器由1点16位构成，可以以16位单位读出、写入。



（c）采用32位指令使用通信寄存器时，连续2个通信寄存器（Wn和Wn＋1）将成为处理对象。顺控程序中指定的通信寄存器编号（Wn）为低位的16位，顺控程序中指定的通信寄存器编号＋1的通信寄存器为高位的16位。例如，采用MOV指令指定W12的情况下，W12为低位的16位；而W13为高位的16位。



采用2个通信寄存器可以存储—2147483648～2147483647，或0H～FFFFFFFFH的数据。

（d）通信寄存器中存储的数据可一直保存到其他数据存入为止。

备 注

MELSECNET/H网络模块内的通信寄存器为16384个，而基本型QCPU内的通信寄存器为2048个。在基本型QCPU上使用2048个以后的通信寄存器时，请在PC参数的软元件设置时更改通信寄存器的个数。

（2）在网络系统上使用时
在网络系统上使用时，必须设置网络参数。
未采用网络参数进行设置的通信寄存器可以作为数据寄存器的代用品使用。

备 注

1）网络参数的有关内容请参照下列手册。
• Q对应MELSECNET/H网络系统参考手册

10.2.14 通信用特殊寄存器（SM）

（1）什么是通信用特殊寄存器
（a）通信用特殊寄存器就是存储MELSECNET/H网络模块等的智能功能模块的通信状态・异常内容的寄存器。
（b）由于数据通信时的信息采用数值形式存储，因此，可以通过对通信用特殊寄存器的监视，调查异常部位和原因。

（2）通信用特殊寄存器的个数
通信用特殊寄存器为SD0～SD3FF的1024个，MELSECNET/H网络模块等的每个智能功能模块带有512个。

备 注

可以使用的通信用特殊寄存器的详细内容请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

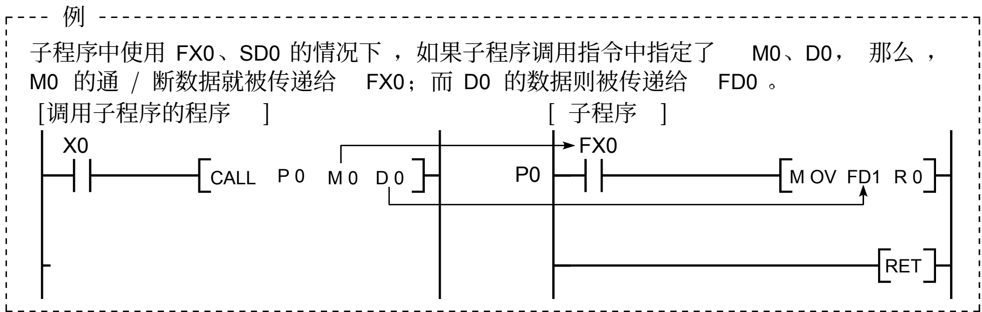
10.3 内部系统软元件

内部系统软元件就是系统所用的软元件。
内部系统软元件的分配和容量是固定的，不可由用户更改。

10.3.1 功能软元件（FX、FY、FD）

（1）什么是功能软元件

(a) 功能软元件就是带自变数的子程序中使用的软元件，用于在调用带参数子程序的程序和带自变数的子程序之间传递数据。



(b) 在子程序中使用功能软元件，就可以决定各个子程序的调用程序中使用的软元件，因此，即使使用同一子程序，也可以不必在意其他子程序的调用程序。

（2）功能软元件的种类

功能软元件有3种，即功能输入（FX）、功能输出（FY）和功能寄存器（FD）。

（a）功能输入（FX）

- 功能输入用于向子程序传递通/断数据的场合。
- 子程序中，接收由带参数的子程序调用指令所指定的位数据，并用于运算。
- 基本型QCPU的位数据设置软元件全部可以使用。

（b）功能输出（FY）

- 功能输出用于将子程序的运算结果（通/断数据）传递给子程序调用程序的场合。
- 运算结果存入带参数的子程序中指定的软元件。
- 可以使用除基本型QCPU的输入（X、DX）以外的位数据指定软元件。

（c）功能寄存器（FD）

• 功能寄存器用于在子程序的调用程序和子程序之间传递数据。
• 功能寄存器的输入/输出条件由基本型QCPU自动进行判别。子程序中为源数据的情况下，就作为子程序的输入数据。
  子程序中为目的数据的情况下，就作为来自子程序的输出数据。
• 每个功能寄存器占用4个字。
  但是，所使用的字数因子程序中的指令而异。
  1个字的指令的情况下，就只使用1个字。



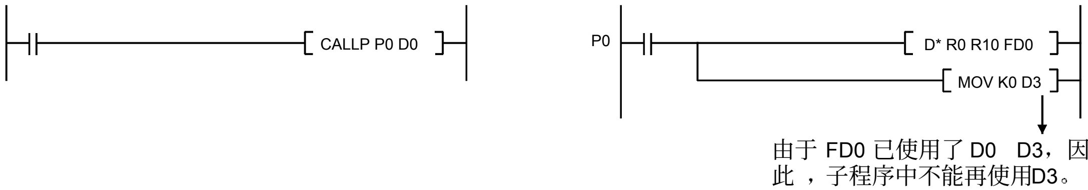
2个字的指令的情况下，就使用2个字。



32位乘除运算的目的地的情况下，使用4个字。



• 在带自变数的字程序内，不可使用功能寄存器已使用的软元件。
  如果将功能寄存器已使用的软元件用在带自变数的子程序中，功能寄存器的值就不能正常地传递给调用程序。



• 可以使用基本型QCPU的字数据指定软元件。

备 注

1）关于功能寄存器的使用方法，请参照下列手册。
  • QCPU（Q模块）/QnACPU编程手册（通用指令篇）

10.3.2 特殊继电器（SM）

（1）什么是特殊继电器

（a）特殊继电器就是存储基本型QCPU状态的继电器。

（2）特殊继电器的分类

特殊继电器根据其用途可分类如下。

（a）故障诊断用　：SM0～SM99
（b）串行通信用　：SM100～SM129
（c）系统信息　：SM200～SM399
（d）系统时钟/系统计数器　：SM400～SM499
（e）扫描信息　：SM500～SM599
（f）存储卡信息　：SM600～SM699
（g）指令相关　：SM700～SM799

备　注

基本型QCPU上可以使用的特殊继电器的详细内容请参照附录1。

10.3.3 特殊寄存器（SD）

（1）什么是特殊寄存器

（a）特殊寄存器就是存储基本型QCPU状态（故障诊断、系统信息等）的寄存器。

（2）特殊寄存器的分类

特殊寄存器根据其用途可分类如下。

（a）故障诊断用　　：SD0～SD99
（b）串行通信功能用　　：SD100～SD129
（c）保险丝断路模块ト　　：SD130～SD149
（d）输入输出模块核对　　：SD150～SD199
（e）系统信息　　：SD200～SD399
（f）系统时钟/系统计数器　　：SD400～SD499
（g）扫描信息　　：SD500～SD599
（h）存储卡信息　　：SD600～SD699
（i）指令相关　　：SD700～SD799

备　注

基本型QCPU上可以使用的特殊寄存器的详细内容请参照附录2。

10.4 直接通信软元件（J□\□）

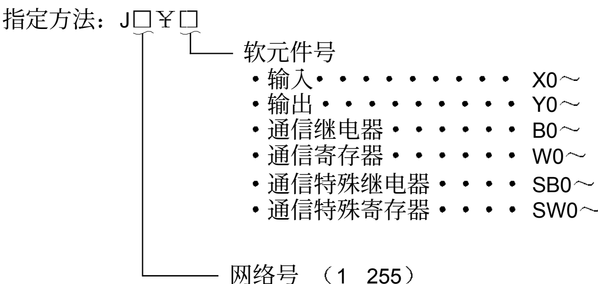
（1）什么是直接通信软元件

（a）基本型QCPU和MELSECNET/H网络系统的网络模块之间的刷新（数据传输）由顺控程序的结束处理进行。

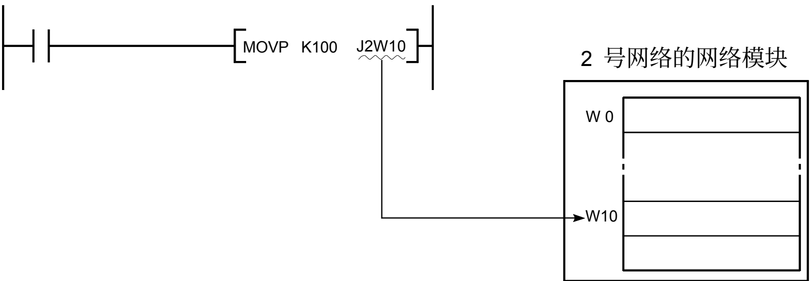
直接通信软元件就是对MELSECNET/H网络模块内的通信软元件直接进行存取的软元件。

（b）指定方法

• 直接通信软元件采用网络号和软元件号指定。



• 2号网络的通信寄存器10（W0）的情况下，就写作“J2¥W10”。



• 位软元件（X、Y、B、SB）的情况下，指定位数。

指定举例：J1¥1X0，J10¥K4B0

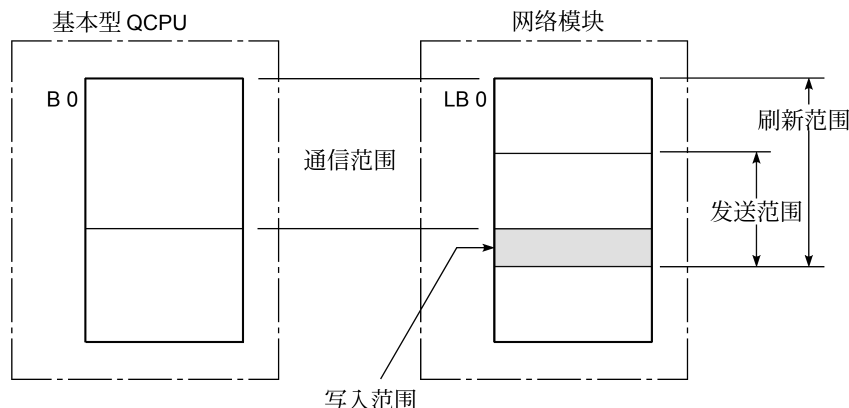
（2）指定范围

直接通信软元件可以指定网络的所有通信软元件。

［网络刷新指定时没有指定范围的通信寄存器也可以指定 。］

（a）写入时

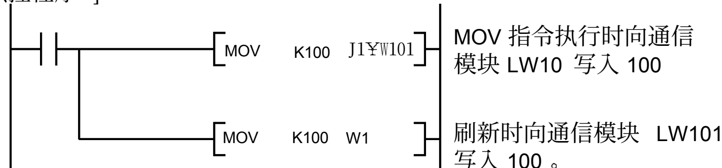
① 请使用网络参数的通用参数设置在发送范围内的通信软元件的范围内，写入到未用网络刷新参数设置在刷新范围内的范围内。



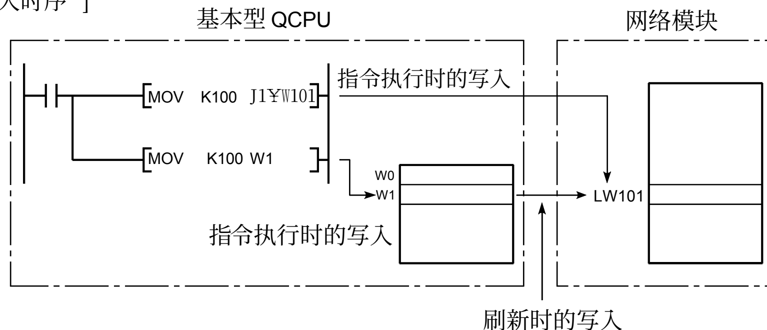
② 虽然也可以写入使用刷新参数设置在刷新范围内的通信软元件范围内，但刷新时通信模块的通信软元件的数据将改写。
因此，采用直接通信软元件进行写入时，刷新参数下所设置的基本型的相应软元件内也请写入同一数据。

〔刷新参数的设置〕
• 网络号：1
• 基本型QCPU（W0～W3F）←→网络模块（LW0～LW3F）

[顺控程序]



[写入时序]



③ 利用直接通信软元件向其他站的写入范围内进行写入的情况下，在从其他站接收数据之时改写成所接收到的数据。

（b）读出时

利用直接通信软元件进行的读出可以在网络模块的通信软元件的整个范围内进行。

（3）直接通信软元件和通信刷新的不同点

直接通信软元件和通信刷新的不同点如下表10.4所示：

表10.4 直接通信软元件和通信刷新的不同点

<table>
<tr><th colspan="2">项　目</th><th>直接通信软元件</th><th>通信刷新</th></tr>
<tr><td rowspan="4">程序中的表达方法</td><td>通信继电器</td><td>J□¥K4B0～</td><td>B0～</td></tr>
<tr><td>通信寄存器</td><td>J□¥W0～</td><td>W0～</td></tr>
<tr><td>通信特殊继电器</td><td>J□¥K4SB0～</td><td>SB0～</td></tr>
<tr><td>通信特殊寄存器</td><td>J□¥SW0～</td><td>SW0～</td></tr>
<tr><td colspan="2">步数</td><td>2步</td><td>1步</td></tr>
<tr><td colspan="2">与网络模块的存取范围</td><td>各网络模块的所有通信软元件</td><td>刷新参数所设置的范围</td></tr>
<tr><td colspan="2">存取数据的保证范围</td><td colspan="2">字（16位）单位</td></tr>
</table>

备　注

1）MELSECNET/H网络系统的有关内容请参照下列手册。
- Q对应MELSECNET/H网络系统参考手册

2）网络参数、通用参数网络刷新参数的有关内容请参照下列手册。
- 详细说明：Q对应MELSECNET/H网络系统参考手册
- 设置方法：GX Developer操作手册

10.5 智能功能模块软元件（U□￥G□）

（1）什么是智能功能模块软元件

（a）智能功能模块软元件就是从基本型QCPU上对主基板和扩展基板上安装的智能功能模块的缓冲存储器直接进行存取的软元件。

（b）智能功能模块软元件由智能功能模块的输入输出编号和缓冲存储器地址来加以指定。

指定方法 ： U□ G□

└→ 缓冲存储器地址（设置范围 十进制数的 0 16383 ）＊1

└→ 智能功能模块的输入输出编号

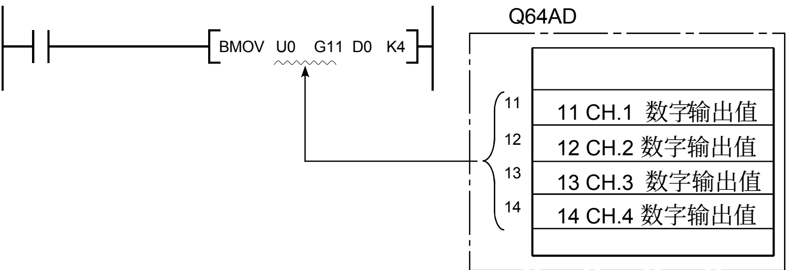
• 设置 采用 3 位表示输入输出编号的情况下为高位的 2 位

X/YF0 的情况下・・・ X/YF0

└ 指定 0F

• 设置范围 Q00JCPU 00H 0FH Q00/Q01CPU 00H 3FH

主基板的插槽0中安装的Q64AD型模/数转换模块（X/Y0～X/Y1F）的CL.1～CH.4的数字输出值存入D0～D3的情况下，指定方法如下图所示。



（2）处理速度

采用智能功能模块软元件的处理速度如下：

（a）对智能功能模块的缓冲存储器进行读出/写入的情况下，通过“FROM/TO指令时的处理速度”，速度多少有所提高。

（例如：“MOV U0￥G11 D0的场合）

（b）采用1条指令从智能功能模块的缓冲存储器读出并进行其他处理的情况下，请以“FROM/TO指令时的处理速度”和“指令的处理速度“的合计值作为参考标准。

（例如：“＋ U0￥G11 D0 D11”的场合）

顺控程序中2次以上使用同一智能功能模块的同一缓冲存储器的数据时，采用FROM指令将数据从智能功能模块的缓冲存储器中读出到基本型的方法可以提高处理速度。

备 注

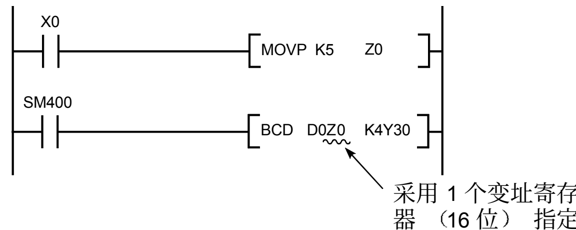
＊1：缓冲存储器的地址、用途的有关内容请参照所使用的智能功能模块的手册。

10.6 变址寄存器（Z）

（1）什么是变址寄存器

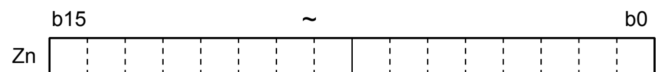
(a) 变址寄存器就是顺控程序中所使用的软元件的间接设置（变址修饰）中使用的软元件。

变址修饰使用1个变址寄存器。



（b）变址寄存器共有10个，即Z0～Z9。

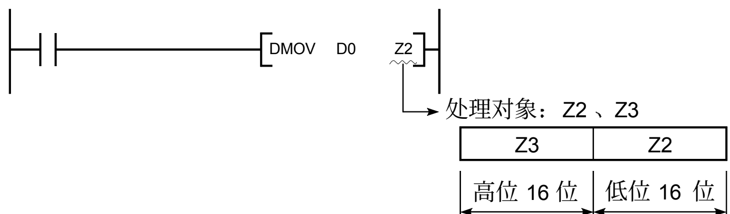
（c）每个变址寄存器由16位构成，可以以16位单位读出、写入。



（d）采用32位指令使用变址寄存器时，Zn和Zn＋1成为处理对象。

顺控程序中指定的变址寄存器编号（Ｚn）为低位的16位；顺控程序中指定的变址寄存器编号＋1的变址寄存器为高位的16位。

例如，DMOV指令中指定Z2的情况下，Z2为低位的16位；而Z3则为高位的16位。



备　注

采用变址寄存器进行变址修饰的有关内容请参照下列手册。

• QCPU（Q模块）QnACPU编程手册（通用指令篇）

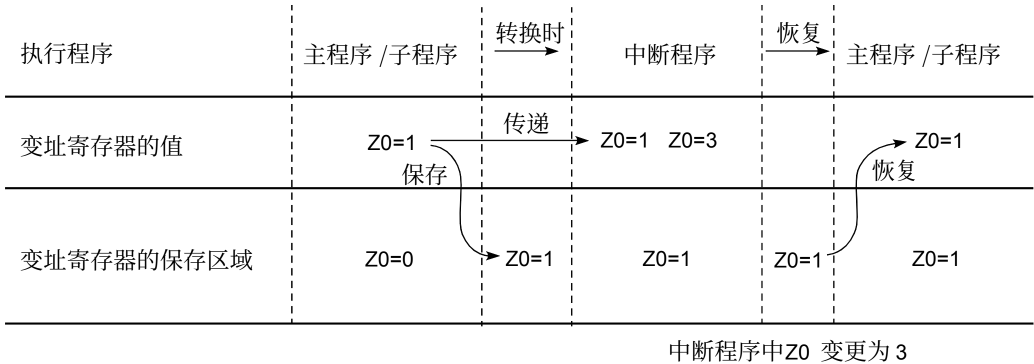
10.6.1 主程序/子程序与中断程序切换时的处理

主程序/子程序和中断程序的切换时变址寄存器（Z0～Z9）的内容是进行保存（保护）•恢复，还是不进行保存，可以利用PC参数的PC系统设置加以选择。
中断程序中不进行变址寄存器写入的情况下，将“中断程序/固定周期程序设置”设置为“高速执行”可以加快转换速度。

（1）不选择高速执行时

（a）从主程序/子程序转切到中断程序时，将主程序/子程序的变址寄存器的值保存后，再传递给中断程序。

（b）从中断程序切换到主程序/子程序时，恢复保存的变址寄存器的值。

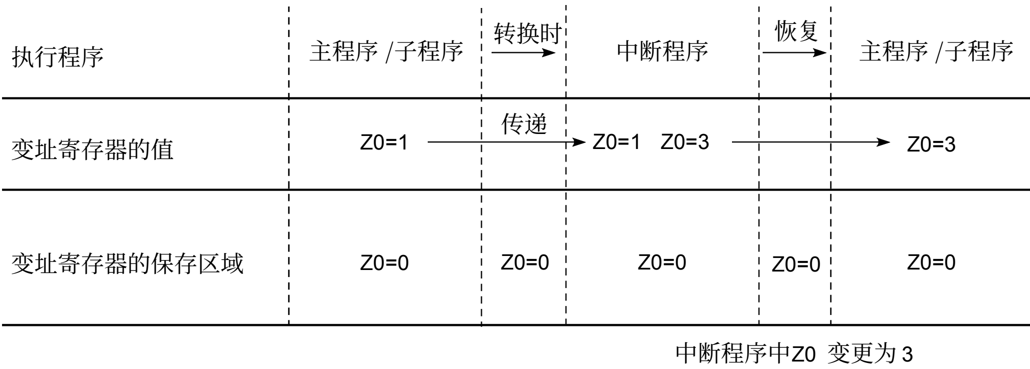


从中断程序向主程序/子程序传递变址寄存器的值时，请使用字软元件。

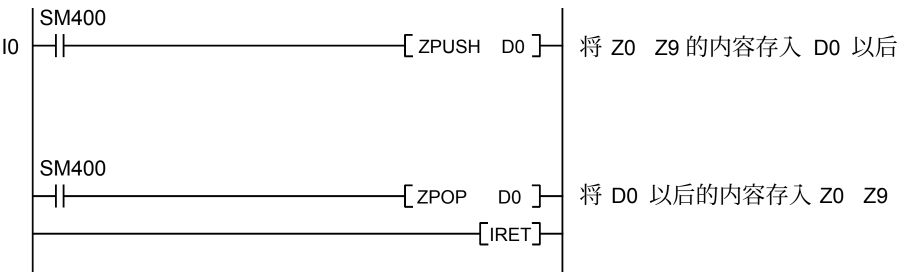
（2）选择高速执行时

（a）从主程序/子程序转换到中断程序之后，不进行变址寄存器的保存和恢复。

（b）如果在中断程序中向变址寄存器写入了数据，主程序/子程序中所使用的变址寄存器的值就被破坏。



（c）中断程序中向变址寄存器写入数据的情况下，请采用ZPUSH指令/ZPOP指令进行变址寄存器的保存和恢复。

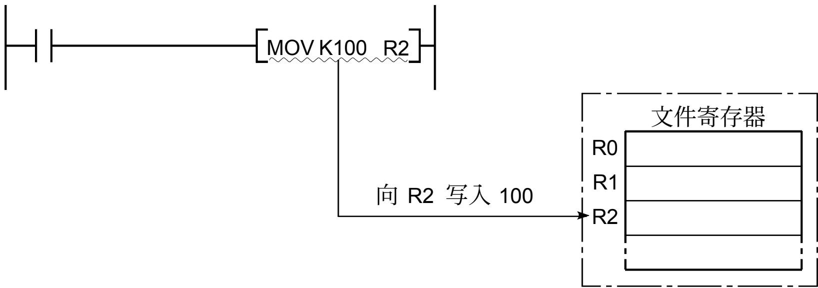


10.7 文件寄存器（R）

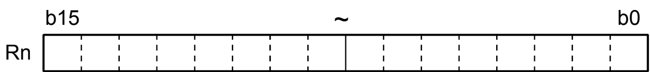
（1）什么是文件寄存器

（a）文件寄存器就是数据寄存器扩展所用的软元件。

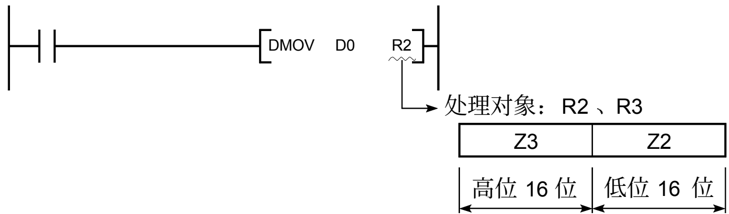
（b）文件寄存器存储在Q00/Q01CPU的标准RAM中。
标准RAM中可以存储32k个文件寄存器，可以和数据寄存器以同样的处理速度使用。



（c）每个文件寄存器由1点16位构成，可以以16位单位读出、写入。



（d）采用32位指令使用文件寄存器时，Zn和Zn＋1成为处理对象。
顺控程序中指定的文件寄存器编号（Rn）为低位的16位；顺控程序中指定的文件寄存器编号＋1的文件寄存器为高位的16位。
例如，DMOV指令中指定R2的情况下，R2 低位的16位；而R3 高位的16位。



采用2个文件寄存器可以存储－2147483648～2147483647，或0H～FFFFFFFFH的数据。

（e）文件寄存器的内容即使进行电源切断/复位操作也能保持。
（即使进行锁存清零也不能初始化。）
电源切断/复位操作时对文件寄存器的内容进行初始化的情况下，请使用顺控程序进行。
例如，可编程控制器的电源启动时对文件寄存器R0～R2047进行清零的情况下，请采用FMOV指令写入“0”。

（f）文件寄存器指定在R0～R32767的范围内指定。
此外，文件寄存器也可以在ZR0～ZR32767的范围内指定。

（2）文件寄存器使用时的注意事项

即使对32k点以上的文件寄存器编号进行写入/读出，也不会出错。
但是，从文件寄存器进行读出时会存入不定的数据，因此，请加以注意。

（3）文件寄存器的删除

文件寄存器可以利用在线的PC数据删除指令进行删除。

10.8 嵌套（N）

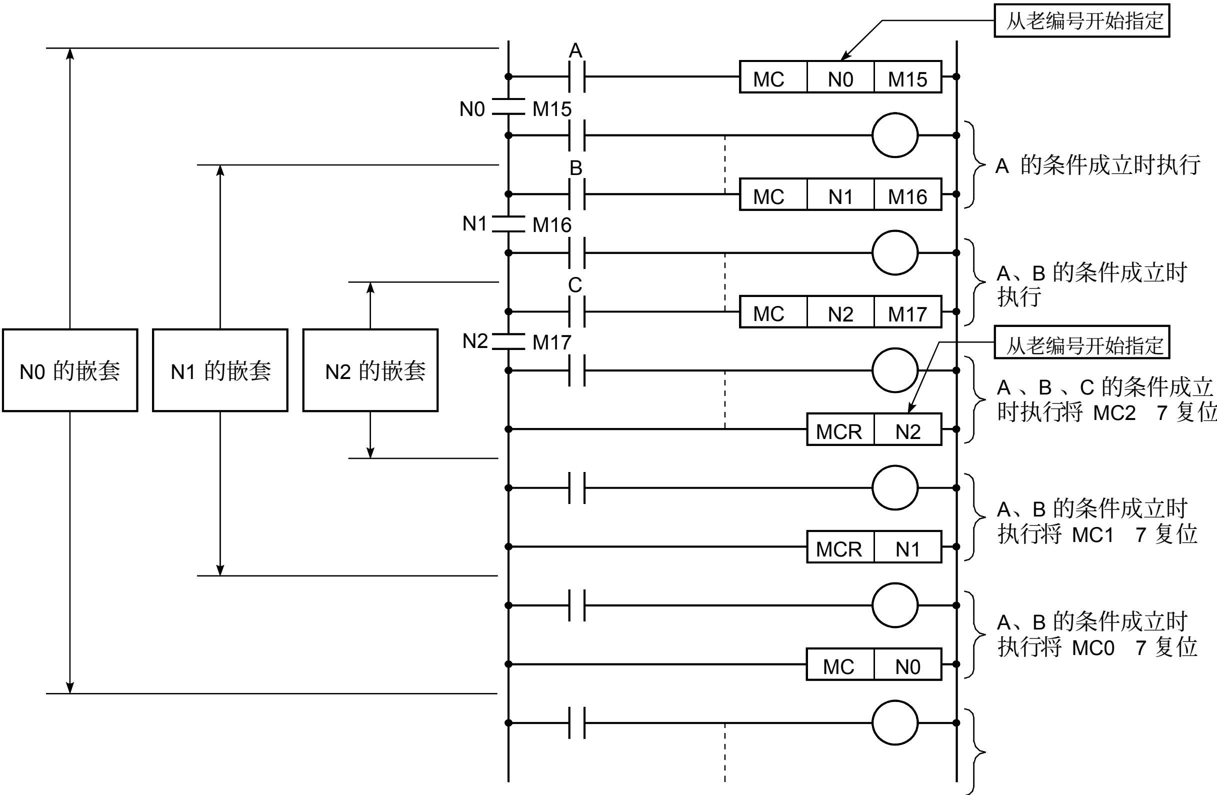
（1）什么是嵌套

嵌套就是采用主控MC指令、MCR指令以嵌套结构形式对动作条件进行编程时所使用的软元件。

（2）主控中的指定方法

主控就是利用电路的通用母线的开关而编制高效率的电路切换顺控程序的指令。
从嵌套结构的外侧开始，从嵌套的小编号（从N0到N7的顺序）起顺序依次指定。
有关使用方法请参照下列手册。

- QCPU（Q模块）/QnACPU编程手册（通用指令篇）



10.9 指针（P）

（1）什么是指针

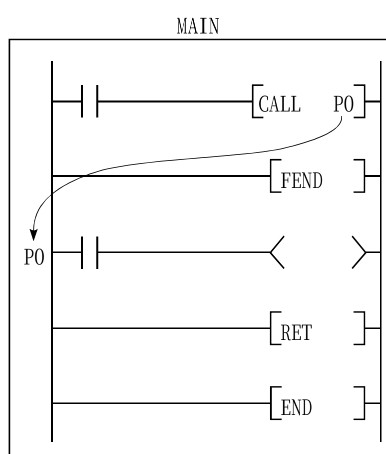
指针就是跳转指令（CJ、SCJ、JUMP）或子程序调用指令（CALL等）所使用的软元件。

指针的点数为300点。

（2）指针的用途

（a）跳转指令（CJ、SCJ、JUMP）的跳转目标的指定和标号（跳转起始位置的指定）

（b）子程序调用指令（CALL、CALLP等）的调用目标和标贴（子程序头部的指定）



备 注

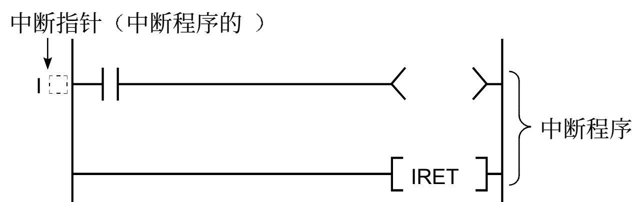
跳转指令、子程序调用指令的有关内容请参照下列手册。

• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

10.10 中断指针（I）

（1）什么是中断指针

（a）中断指针就是作为标贴用在中断程序头部的软元件。



（b）中断指针有128（I0～I127）个。
可以使用的中断指针编号请参照表10.5。

（2）中断指针编号和中断原因

（a）中断指针相对应的中断原因有以下2种。

- I0～I15……………………………来自QI60型中断模块的中断输入。
（由QI60引起的中断）
- I28～I31………………………… 由基本型QCPU的内部定时器引起的固定周期中断。
（由内部定时器引起的中断）

（b）中断指针编号和中断原因一览如表10.5所示。

表10.5 中断指针编号和中断原因一览表

中断号	中断原因		优先顺序
I0	由QI60引起的中断	第1点	1
I1		第2点	2
I2		第3点	3
I3		第4点	4
I4		第5点	5
I5		第6点	6
I6		第7点	7
I7		第8点	8
I8		第9点	9
I9		第10点	10
I10		第11点	11
I11		第12点	12
I12		第13点	13
I13		第14点	14
I14		第15点	15
I15		第16点	16
I16～I27	不可使用	—	—
I28	由内部定时器引起的中断 ＊1	100ms	20
I29		40ms	19
I30		20ms	18
I31		10ms	17
I32～I127	不可使用	—	—

备　注

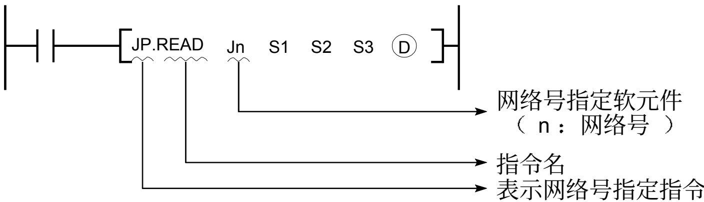
＊1：内部定时器的时限表示的是默认值。
PC参数的PC系统设置时可以在2 ms～1000 ms的范围内以1 ms单位变化。

10.11 其他软元件

10.11.1 网络号指定软元件（J）

（1）什么是网络号指定软元件
网络号指定软元件就是采用数据通信指令指定网络号时所使用的软元件。

（2）网络号指定软元件的指定方法
网络号指定软元件采用数据通信指令按如下方法指定。



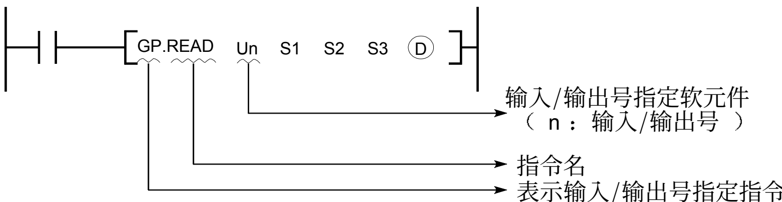
备　　注

数据通信指令的详细内容请参照下列手册。
- Q对应MELSECNET/H网络系统参考手册

10.11.2 输入/输出号指定软元件（U）

（1）什么是输入/输出 号指定软元件
输入/输出号指定软元件就是采用智能功能模块专用指令指定输入/输出号时所使用的软元件。

（2）输入/输出号指定软元件的指定方法
输入/输出号指定软元件采用智能功能模块专用指令按如下方法指定。



备　注

智能功能模块专用指令的详细内容请参照所使用的智能功能模块的手册。

10.11.3 宏指令自变数软元件（VD）

（1）什么是宏指令自变数软元件

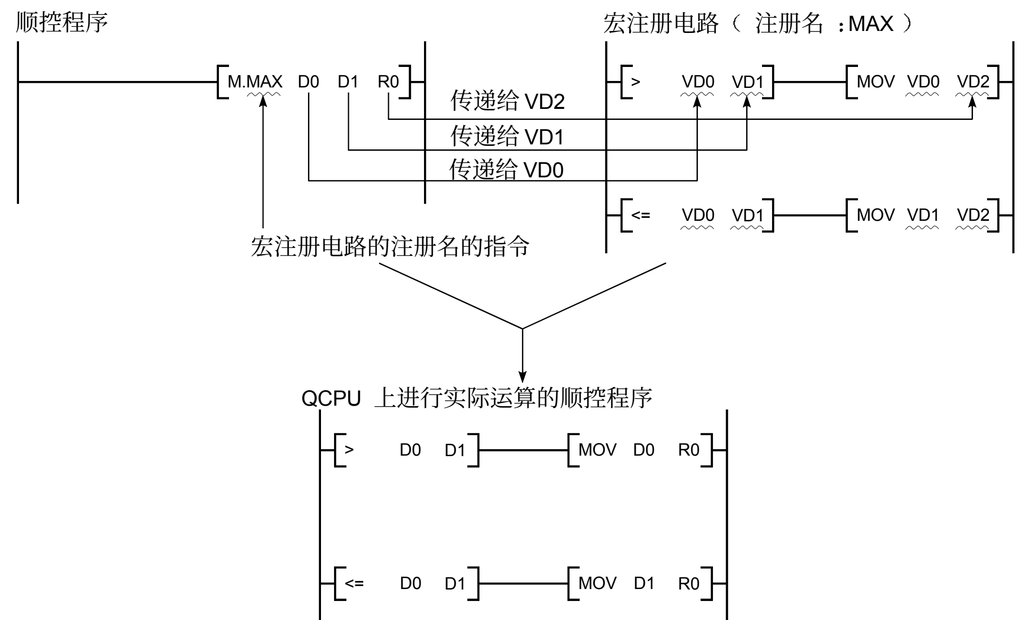
宏指令自变数软元件就是宏注册电路中使用的软元件。

如果在宏注册电路中使用VD□，就变更为使用宏指令时所指定的软元件。

（2）宏指令自变数软元件的指定方法

宏指令自变数软元件指定GX Developer宏注册时作为宏指令注册的电路中使用的软元件中设为VD的软元件。*

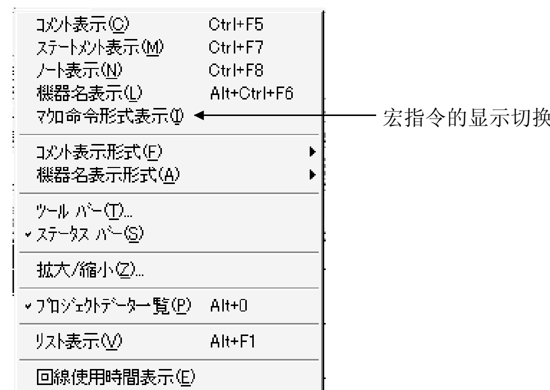
顺控程序中使用宏指令的情况下，指定宏注册电路中使用的对应于宏指令自变数软元件的新编号顺序的软元件。



备　注

1）＊：宏指令自变数软元件在1个宏注册电路中可以使用VD0～VD9。

2）GX Developer上的读出模式下的程序表示时，也可以采用宏指令的形式。（显示的转换利用“宏指令形式显示”进行。）



10.12 常数

10.12.1 十进制常数

（1）什么是十进制常数
十进制常数就是顺控程序中指定十进制数的软元件。
K□形式（例如：K1234）指定，在基本型QCPU内部以二进制数（BIN）形式存储。
有关二进制数（BIN）的详细内容请参照4.8.1项。

（2）指定范围
十进制常数的设置范围如下。
- 使用字数据（16位）时…………K—32768～K32767
- 使用双字数据（32位）时………K—2147483648～K2147483647

10.12.2 十六进制常数

（1）什么是十六进制常数
十六进制常数就是顺控程序中指定十六进制数或BCD数据的软元件。
（采用BCD码指定数据时，用0～9指定十六进制数的各个数据位。）
采用H□形式（例如：H1234）指定。
有关十六进制数的详细内容请参照4.8.2项。

（2）指定范围
十六进制常数的设置范围如下：
- 使用字数据（16位）时……………H0～HFFFF（BCD数据时为H0～H9999）
- 使用双字数据（32位）时…………H0～HFFFFFFFF
（BCD数据时为H0～H99999999）

10.12.3 字符串（“　　”）

（1）什么是字符串常数
字符串常数就是顺控程序中指定字符串的软元件。
用“　　”筐起半角字符的形式（例如：“ABCD1234”）指定。

（2）可以使用的字符
字符串中可以使用JIS8编码。
基本型QCPU对大写字母和小写字母进行区分。

（3）指定字符数
字符串从指定字符开始到NUL编码（00H）为止计算字符单位。
但是，使用字符串的指令（$MOV指令）可以指定的字符串最大不超过32字符。

要　　点
字符串只可在$MOV指令中使用。

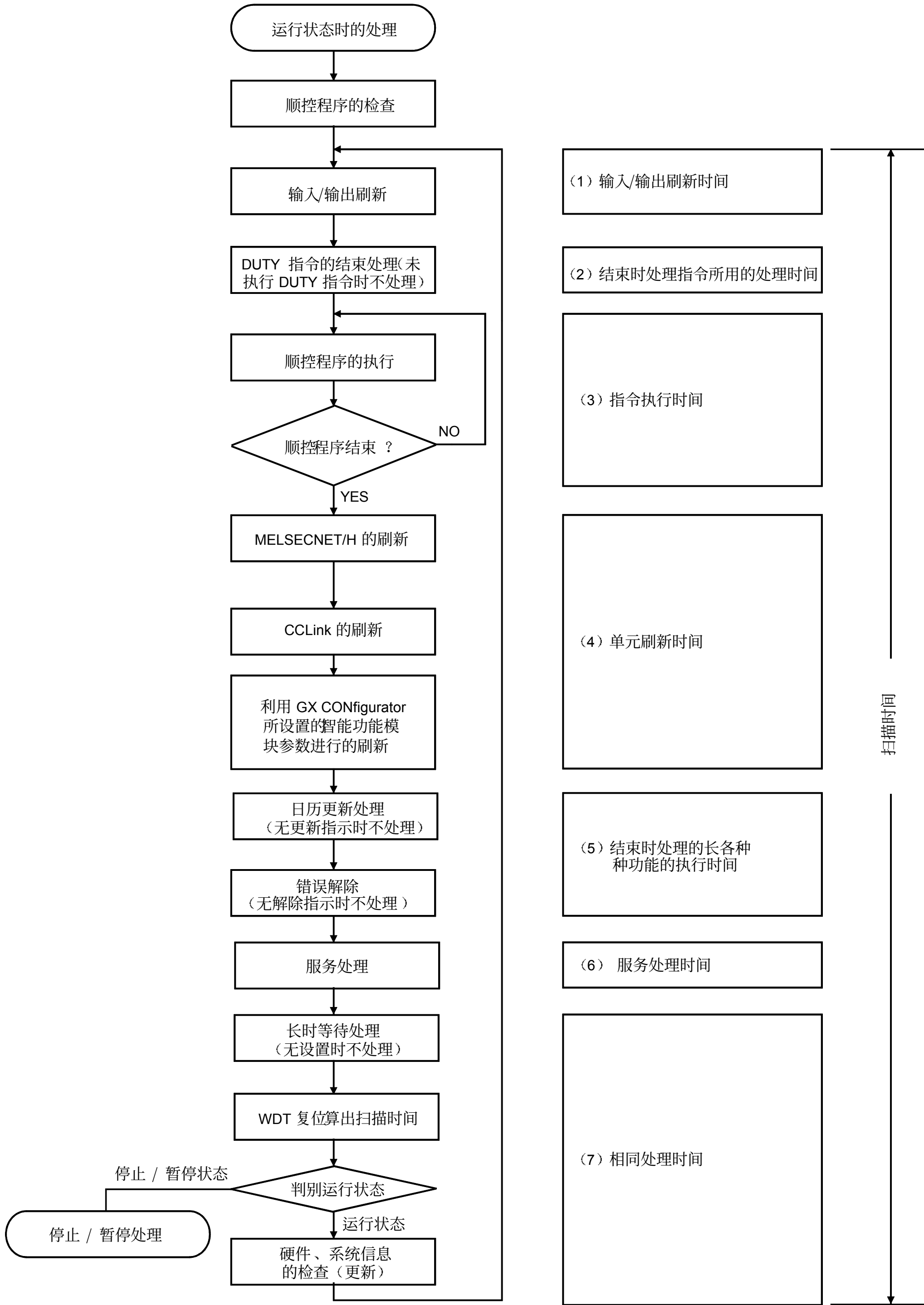
11

第11章 基本型QCPU的处理时间

本章阐述关于基本型QCPU的处理时间的分析。

11.1 扫描时间的构成

基本型QCPU处于运行状态时，周而复始地进行以下的处理。



（）内的数字为11.2节中的项目号。

11.2 扫描时间的分析

扫描时间根据以下因素变化。
- 输入输出点数
- 1次扫描内执行的所有指令的处理时间
- 1次扫描内执行的用户中断程序的合计处理时间
- 结束时处理的用于指令的处理
- 结束时处理的各种功能的执行
- 模块刷新（由MELSECNET/H、CC—Link引起的刷新等）
- 服务处理的处理内容
- 恒定扫描设置（用参数设置）

扫描时间为以下的处理时间的合计值。

（1）输入/输出刷新时间

（a）和基本型QCPU的主基板、扩展基板上安装的以下模块进行输入输出数据刷新的时间。
- 输入模块
- 输出模块
- 智能功能模块

（b）输入/输出刷新时间采用下式算出。

（输入/输出刷新时间）＝（输入点数/16）×N1＋（输出点数/16）×N2

N1、N2请参照下表。

CPU类型	N1		N2	
	Q3□B	Q6□B	Q3□B	Q6□B
Q00JCPU	2.5μs	3.3μs	1.3μs	2.3μs
Q00CPU	2.4μs	3.2μs	1.3μs	2.3μs
Q01CPU	2.3μs	3.1μs	1.3μs	2.3μs

（2）结束时处理的用于指令的处理时间

（a）DUTY指令

DUTY指令中指定的用户计时时钟（SM420～SM424）进行结束处理时的通/断时间。

CPU类型	结束处理时间	
Q00JCPU	0.15ms	0.21ms
Q00CPU	0.14ms	0.19ms
Q01CPU	0.12ms	0.16ms

（3）指令执行时间

（a）指令执行时间是基本型QCPU上执行的程序中使用的各指令的处理时间的合计值。□
关于各条指令的处理时间，请参照下列手册。
• QCPU（Q模块）/QnACPU编程手册（通用指令篇）

（b）中断程序包括中断程序起动前辅助操作时间和中断程序结束辅助操作时间。
指令执行时间中请加上中断程序起动前辅助操作时间和中断程序结束辅助操作时间。

① 中断程序起动前辅助操作时间（BI）

CPU类型	固定周期中断处理（I28～I31）		来自QI60（I0～I15）的中断处理＊1	
	无高速起动	有高速起动	无高速起动	有高速起动
Q00JCPU	175μs	150μs	350μs	325μs
Q00CPU	145μs	125μs	285μs	265μs
Q01CPU	135μs	120μs	270μs	255μs

＊1：QI60表示的是安装在主基板上的插槽0中的情况下的值。

② 中断程序的结束辅助操作时间（B2）

CPU类型	无高速起动	有高速起动
Q00JCPU	175μs	150μs
Q00CPU	145μs	125μs
Q01CPU	135μs	120μs

（4）模块刷新时间

（a）MELSECNET/H的刷新
基本型QCPU和MELSECNET/H网络模块之间的刷新时间。
关于MELSECNET/H的刷新时间，请参照下列手册。
• Q对应MELSECNET/H网络系统参考手册

（b）CC—Link的自动刷新
基本型QCPU和CC—Link的主站•本地站模块之间的刷新时间。
关于CC—Link的自动刷新时间，请参照下列手册
• QJ61BT11型CC—Link系统主站•本地站模块用户手册（详细篇）

（c）智能实用包（智能自动刷新）
① 根据智能功能模块实用包所设置的自动刷新设置，相应智能功能模块和CPU模块之间刷新时间。

② 智能自动刷新时间可根据下式进行计算。
（刷新时间）＝KN1＋KN2×（刷新点数）

③ KN1、KN2采用下表的值。
主基板上安装智能模块时

CPU类型	KN1	KN2
Q00JCPU	111μs	55μs
Q00CPU	91μs	46μs
Q01CPU	85μs	41μs

扩展基板上安装智能模块时

CPU类型	KN1	KN2
Q00JCPU	113μs	56μs
Q00CPU	92μs	48μs
Q01CPU	86μs	43μs

（例）
针对模/数转换模块（Q64AD）的自动刷新点数为4点的场合（安装在Q01CPU的主基板上时）
0.249（ms）＝0.085＋0.041×4

（5）结束时处理的各种功能的执行时间
（a）日历更新处理时间
① 时钟数据设置请求(SM210由断变通)时,结束处理时用于将存储在SD210～SD213中的时钟数据写入时钟因素的时间。
② 时钟数据读出请求（SM213接通）时，结束处理时用于读出时钟数据到SD210～SD213中的时间。

CPU类型	结束处理时间	
	时钟数据设置请求时	时钟数据设置请求时
Q00JCPU	0.12ms	0.04ms
Q00CPU	0.11ms	0.03ms
Q01CPU	0.10ms	0.02ms

（b）出错解除时间
SM50（出错解除）的上升（由断变通）时，SD50中存储的继续出错的出错解除时间。

CPU类型	通用处理时间	
	信号报警器	信号报警器
Q00JCPU	0.17ms	0.10ms
Q00CPU	0.14ms	0.09ms
Q01CPU	0.13ms	0.08ms

(6) 服务处理时间

(a) 采用GX Developer的监视

在GX Developer上进行监视的情况下的处理时间。

GX Developer上进行监视的情况下须加上。

功 能	连接本地站CPU模块的RS—232时			连接其他站时＊4		
	Q00JCPU	Q00CPU	Q01CPU	Q00JCPU	Q00CPU	Q01CPU
程序的PC读出＊1	1.6ms	1.3ms	1.2ms	2.3ms	1.9ms	1.8ms
软元件监视＊2	1.2ms	1.0ms	0.9ms	2.4ms	2.0ms	1.9ms
运行中写入＊3	1.0ms	1.0ms	1.0ms	1.9ms	1.6ms	1.5ms

＊1：从程序存储器中读出8 k步的程序时花费的时间。

＊2：注册监视中设置32点时花费的时间。

＊3：追加100步的电路时花费的时间。

＊4：表示通过MELSECNET/H、Ethernet、CC—Link、串行通信模块存取时。

(b) 和串行通信模块、Ethernet接口模块的通信

用于和串行通信模块、Ethernet接口模块进行通信的时间。

关于和各种模块的通信时间，请参照下列手册。

• Q对应MELSEC通信协议参考手册

(7) 相同处理

系统所处理的CPU模块的相同处理。

相同处理时间的值如下表所示。

	CPU类型		
	Q00JCPU	Q00CPU	Q01CPU
相同处理时间	0.70ms	0.55ms	0.50ms

上表中的时间是不使用恒定扫描功能时的时间。

使用恒定扫描功能的情况下，常数扫描设置的不足部分进行等待处理。

11.3 其他处理时间

(1) 恒定扫描的精度

CPU类型	无监视，无用户中断	有监视，无用户中断	无监视，有用户中断	有监视，有用户中断
Q00JCPU	0.20ms	0.90ms	中断程序的执行时间（参照11.2节（3）之（b））	以下的合计时间 ① 边“有监视，无用户中断”栏所示的时间 ② 中断程序的合计执行时间
Q00CPU	0.12ms	0.60ms		
Q01CPU	0.10ms	0.50ms		

有监视：表示连接GX Developer进行监视的状态或利用串行通信功能和外部设备进行通信的状态。

无监视：表示不利用GX Developer和串行通信功能进行通信的状态。

第12章　程序写入基本型QCPU之前的步骤

本章阐述GX Developer上编写的程序写入基本型QCPU时的步骤。

12.1　编写程序时的研讨内容

在基本型QCPU上编写程序时，必须事先决定程序容量和所使用的软元件数等。

（1）程序容量的研讨
研讨是否可以在所使用的CPU上可以执行的程序容量以内存储程序和参数。
各CPU上可以执行的程序容量如下所示。
- Q00JCPU　：8k步
- Q00CPU　：8k步
- Q01CPU　：14k步

（2）软元件的用途和点数的设置
研讨程序中使用的软元件的用途和点数。
关于基本型QCPU上可以使用的软元件，请参照第10章。

（3）ROM运行的研讨
ROM运行的情况下，进行PC参数的引导文件设置。

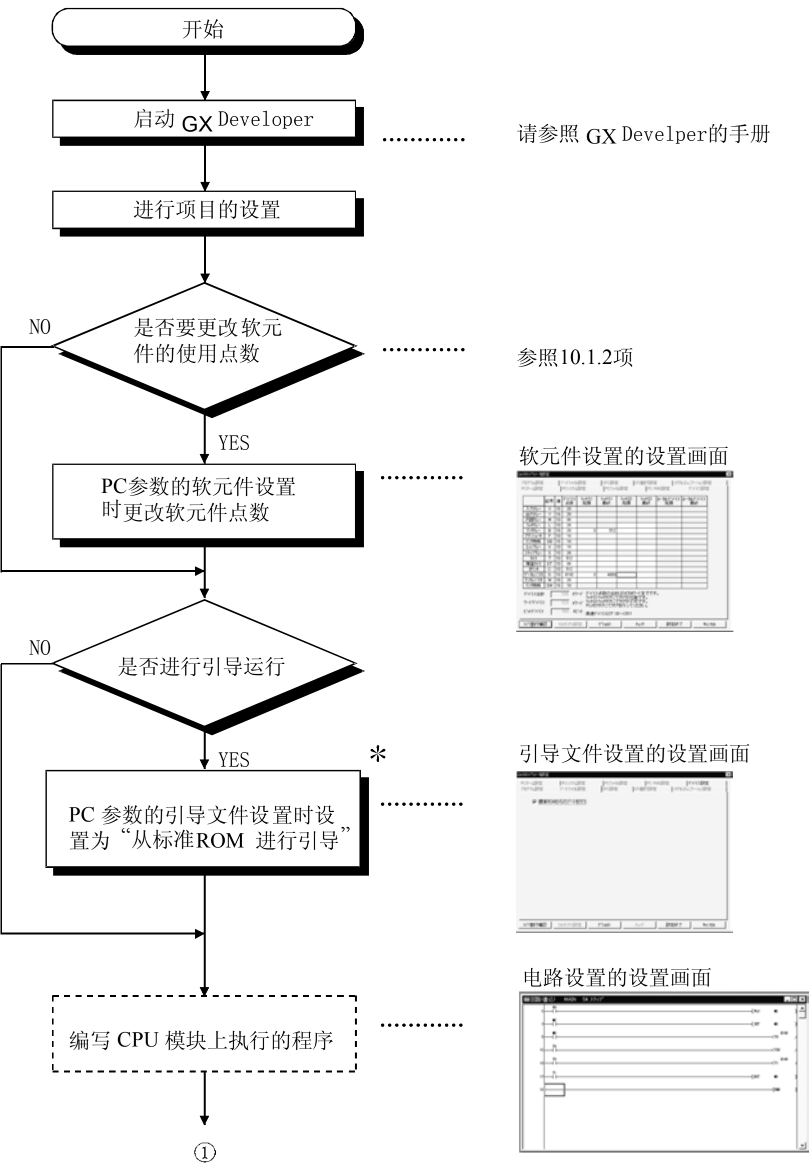
12

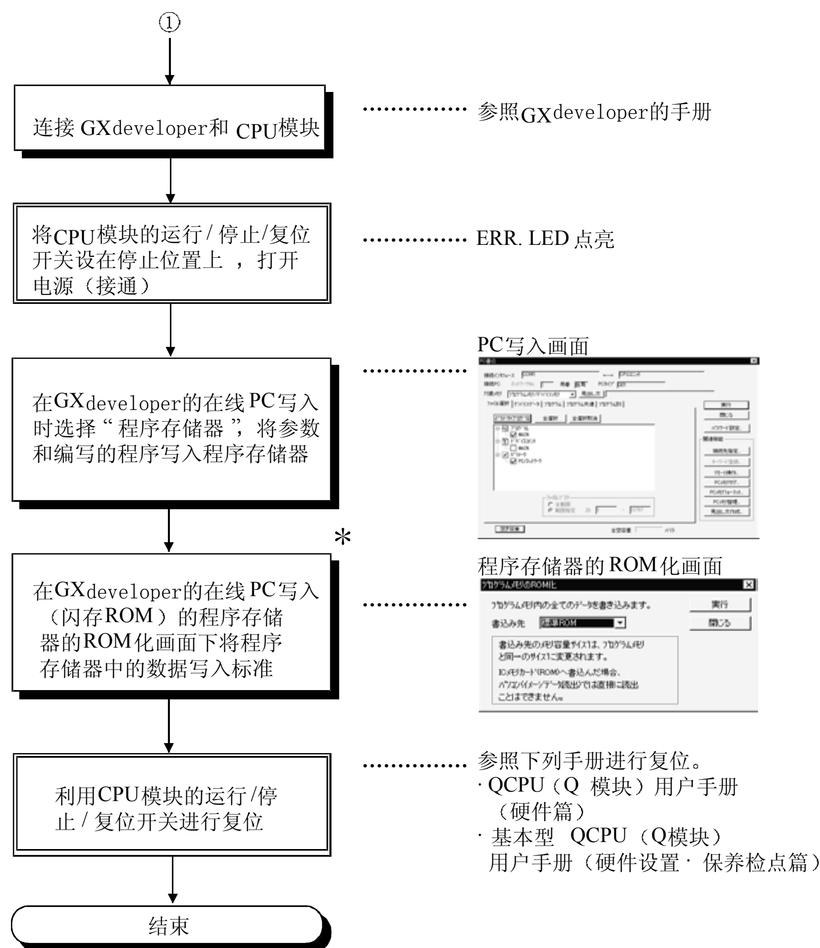
12.2 程序写入基本型QCPU完成之前的步骤

以下所示为GX Developer上编写的参数、程序写入基本型QCPU的标准ROM时的步骤。

参数、程序写入基本型QCPU的标准ROM时，不需要以下步骤中带“＊”的操作。

在以下步骤中，“▭”表示GX Developer一侧的操作项目；而“▭”表示 基本型QCPU一侧的操作项目。




①
连接 GXdeveloper和 CPU模块
参照GXdeveloper的手册
将CPU模块的运行/停止/复位开关设在停止位置上 ，打开电源（接通）
ERR. LED 点亮
在GXdeveloper的在线PC写入时选择“程序存储器”，将参数和编写的程序写入程序存储器
PC写入画面
*
在GXdeveloper的在线PC写入（闪存ROM）的程序存储器的ROM化画面下将程序存储器中的数据写入标准
程序存储器的ROM化画面
利用CPU模块的运行/停止/复位开关进行复位
参照下列手册进行复位。
· QCPU（Q 模块）用户手册（硬件篇）
· 基本型 QCPU（Q模块）用户手册（硬件设置· 保养检点篇）
结束

附录

附录1 特殊继电器一览

特殊继电器SM是可编程控制器内部已确定技术规范的内部继电器。因此，在顺控程序中，不可象通常的内部继电器一样使用。但是，根据需要，可以利用特殊继电器的通/断对CPU模块进行控制。

一览表的各个项目的理解重点如下。

项目	项目说明
编号	• 表示特殊继电器的编号。
名称	• 表示特殊继电器的名称。
内容	• 表示特殊继电器的有关内容。
详细内容	• 对特殊继电器的有关内容作出详细说明。
设置侧 （设置时期）	• 对设置侧和系统侧设置情况下的时期进行说明。 <设置侧> S ：由系统侧设置。 U ：由用户侧设置（来自顺控程序或外围设备的测试操作）。 S/U ：由系统/用户双方设置。 <设置时期>→仅当系统侧设置时，表示设置时期。 每次结束 ：每次结束处理时设置。 初期 ：仅在初始化（电源接通、停止→运行等）时设置。 状态变化 ：发生状态变化时设置。 发生出错 ：发生出错时设置。 执行指令 ：执行指令时设置。 请求时 ：仅在用户（利用SM等）发出请求时设置。

以下项目的详细内容请参照下列手册。

• 网络相关→ • Q对应MELSECNET/H网络系统参考手册（PC间网络篇）

特殊继电器一览

(1) 诊断信息

编号	名称	内容	详细内容	设置侧 (设置时期)
SM0	诊断出错	断：无出错 通：有出错	• 诊断的结果，发生出错就接通。（也包括信号报警器通时。） • 此后即使正常也保持接通。	S（发生出错）
SM1	自我诊断出错	断：无自我诊断出错 通：有自我诊断出错	• 自我诊断的结果，发生出错就接通。（不包括信号报警器接通时。） • 此后即使正常也保持接通。	S（发生出错）
SM5	通用出错信息	断：无通用出错信息 通：有通用出错信息	• SM0接通时，有通用出错信息就接通。	S（发生出错）
SM16	各别出错信息	断：无各别出错信息 通：有各别出错信息	• SM0接通时，有各别出错信息就接通。	S（发生出错）
SM50	出错解除	断→通：出错解除	• 进行出错解除动作。	U
SM51	电池低锁存	断：正常 通：电池低下	• CPU的电池电压低于规定时接通。 • 此后即使电压恢复正常也保持接通。	S（发生出错）
SM52	电池低下	断：正常 通：电池低下	• 和SM51相同，但此后电压恢复正长时切断。	S（发生出错）
SM53	检出AC/DC DOWN	断：无AC/DC DOWN 通：有AC/DC DOWN	• 使用AC电源模块时发生20 ms以上的瞬间掉电时接通，电源断→通时复位。 • 使用DC电源模块时发生10 ms以上的瞬间掉电时接通，电源断→通时复位。	S（发生出错）
SM56	运算出错	断：正常 通：有运算出错	• 发生运算出错时接通。 • 此后即使恢复正常也保持接通。	S（发生出错）
SM60	检出保险丝短路	断：正常 通：有保险丝断路模块	• 只要有1个保险丝处于断路状态的输出模块就接通，此后即使恢复正常也保持接通。	S（发生出错）
SM61	输入输出模块 核对出错	断：正常 通：有出错	• 如果和输入输出模块电源打开时所注册的状态不同就接通，此后即使恢复正常也保持接通。	S（发生出错）
SM62	检出信号报警器	断：未检出 通：检出	• 只要有1个信号报警器标志接通就接通。	S（发生出错）
SM100	串行通信功能使用的标志	断：不使用串行通信功能 通：使用串行通信功能	• 存储串行通信功能设置的使用/不使用存储串行通信功能。	S（电源接通、复位时）
SM101	通信协议状态 标志	断：GX Developer 通：MC通信协议元件	• 存储利用RS—232通信的设备是GX Developer还是MC协议通信设备。	S（RS—232通信时）
SM110	协议异常	断：正常 通：异常	• 串行通信功能中采用异常的协议进行通信时接通。 • 此后即使恢复正常也保持接通。	S（发生出错）
SM111	通信状态	断：正常 通：异常	• 串行通信功能中采用不同于设置的模式进行通信时接通。 • 此后即使恢复正常也保持接通。	S（发生出错）
SM112	出错信息清零	通时清零	• 清除SM110、SM111和SM110、SM111中存储的出错码时接通。	U
SM113	溢出出错	断：正常 通：异常	• 串行通信功能中发生溢出出错时接通。	S（发生出错）
SM114	奇偶出错	断：正常 通：异常	• 串行通信功能中发生奇偶出错时接通。	S（发生出错）
SM115	成帧误差	断：正常 通：异常	• 通信功能中发生成帧误差时接通。	S（发生出错）

特殊继电器一览

(2) 系统信息

编号	名称	内容	详细内容	设置侧（设置时期）
SM203	停止触点	停止状态	• 处于停止状态时接通。	S（状态变化）
SM204	暂停触点	暂停状态	• 处于暂停状态时接通。	S（状态变化）
SM206	暂停允许线圈	断：禁止暂停 通：允许暂停	• 远程暂停触点接通时，如果本继电器接通，就进入暂停状态。	U
SM210	时钟数据设置请求	断：无处理 通：有设置请求	• 本继电器由断→通变化的扫描的END指令执行后 ，将SD210～SD213中存储的时钟数据写入时钟因素。	U
SM211	时钟数据出错	断：无出错 通：有出错	• 时钟数据（SD210～SD213）的值发生出错时接通，无出错时切断。	S（请求时）
SM213	时钟数据读请求	断：无处理 通：读出请求	• 本继电器接通时，将时钟数据以BCD值的形式读出到SD210～SD213。	U
SM315	通信确保时间的时间等待有效/无效标志	断：不等待时间 通：等待时间	• SD315中设置了通信处理确保时间时变为有效的标志 • 即使没有通信处理，要求结束处理中等待SD315中设置的时间时也接通。（扫描时间延长，延长的时间就是SD315中设置的时间。） • 没有通信处理时，不要求等待SD315中设置的时间时切断。（默认为断）	U

(3) 系统时钟/计数器

编号	名称	内容	详细内容	设置侧（设置时期）
SM400	常闭	ON OFF	• 常闭。	S（每次结束）
SM401	常开	ON OFF	• 常开。	S（每次结束）
SM402	仅运行后1次扫描时接通	ON OFF 1 次扫描	• 运行后，仅1次扫描时接通。	S（每次结束）
SM403	仅运行后1次扫描切断	ON OFF 1 次扫描	• 运行后，仅1次扫描切断	S（每次结束）
SM410	0.1秒时钟	0.05秒 0.05秒	• 按指定时间反复通/断。 • 电源切断或复位时，从断开始。 * 即使在程序执行途中，一旦到达指定时间，就会产生通—断的状态变化，请加以注意。	S（状态变化）
SM411	0.2秒时钟	0.1秒 0.1秒		
SM412	1秒时钟	0.5秒 0.5秒		
SM413	2秒时钟	1秒 1秒		
SM414	2n秒时钟	n秒 n秒	• 依照SD414所指定的秒数反复通/断。	S（状态变化）
SM420	0号用户计时时钟	n2 扫描 n1 扫描 n2 扫描	• 按一定间隔反复通/断的继电器。 • 电源接通或复位时从断开始。 • 采用DUTY指令设置通/断的间隔。 DUTY n1 n2 SM420	S（每次结束）
SM421	1号用户计时时钟			
SM422	2号用户计时时钟			
SM423	3号用户计时时钟			
SM424	4号用户计时时钟			

(4) 存储卡

编号	名称	内容	详细内容	设置侧 （设置时期）
SM620	存储卡B 可以使用标志	断：不可使用 通：可以使用	• 常闭。	S（开始时）
SM621	存储卡B 保护标志	断：无保护 通：有保护	• 常闭。	S（开始时）
SM622	驱动器3标志	断：无驱动器3 通：有驱动器3	• 常闭。	S（开始时）
SM623	驱动器4标志	断：无驱动器4 通：有驱动器4	• 常闭。	S（开始时）
SM640	使用文件寄存器	断：文件寄存器未使用 通：文件寄存器使用中	• 文件寄存器使用期间接通。（仅限Q00CPU、Q01CPU）	S（状态变化）
SM660	引导运行	断：程序存储器执行 通：引导运行中	• 引导运行期间接通。	S（状态变化）

(5) 指令相关

编号	名称	内容	详细内容	设置侧 （设置时期）
SM700	进位标志	断：进位断 通：进位通	• 用于应用指令中的进位标志。	S（指令执行时）
SM702	查找方法	断：顺序查找 通：折半查找	• 指定查找指令中的查找方法。 • 折半查找时，数据必须已排序。	U
SM703	排序顺序	断：上升序 通：下降序	• 指定排序指令的数据排列方法是升序还是降序。	U
SM704	块比较	断：有不一致 通：完全一致	• BKCMP指令中的所有数据条件成立时接通。	S（指令执行时）
SM715	EI标志	0：中断禁止中 1：中断允许中	• EI指令执行时接通。	S（指令执行时）
SM722	BIN、DBIN指令出错不可标志	断：检查出错 通：不检查出错	• 不希望BIN、DBIN指令中出现“OPERATION ERROR”时接通。	U
SM775	COM指令执行时 通信刷新处理选择	断：进行通信刷新 通：不进行通信刷新	• COM指令执行时只进行一般数据处理的情况下，选择是否进行刷新处理。	U

附录2 特殊寄存器一览

特殊继电器SD是可编程控制器内部已确定的技术规范的内部寄存器。因此，在顺控程序中，不可和通常的内部寄存器一样使用。但是，根据需要，可以利用特殊继电器的通/断对CPU模块进行控制。
特殊寄存器中存储的数据只要没有特别指定，就以BIN值的形式存储。

一览表的各个项目的理解重点如下。

项目	项目说明
编号	• 表示特殊寄存器的编号。
名称	• 表示特殊寄存器的名称。
内容	• 表示特殊寄存器的有关内容。
详细内容	• 对特殊寄存器的有关内容作出详细说明。
设置侧 （设置时期）	• 对设置侧和系统侧设置情况下的时期进行说明。 <设置侧> S ：由系统侧设置。 U ：由用户侧设置（来自顺控程序或GX Developer等的测试操作）。 S/U ：由系统/用户双方设置。 <设置时期>→仅当系统侧设置时，表示设置时期。 每次结束 ：每次结束处理时设置。 初期 ：仅在初始化（电源接通、停止→运行等）时设置。 状态变化 ：发生状态变化时设置。 发生出错 ：发生出错时设置。 执行指令 ：执行指令时设置。 请求时 ：仅在用户（利用SM等）发出请求时设置。

以下项目的详细内容请参照下列手册。
• 网络相关→ • Q对应MELSECNET/H网络系统参考手册（PC间网络篇）

(1) 诊断信息

编号	名称	内容	详细内容	设置侧 （设置时期）
SD0	诊断出错	诊断出错编号	• 将诊断中发生出错时的编号以BIN码形式存储。 • 和故障记录的最新信息内容相同。	S（发生出错）
SD1	诊断出错 发生时刻	诊断出错 发生时刻	• 将SD0数据更新的年份（公历末尾2位）、月份以2位BCD码的形式存储。 B15 ～ B8 B7 ～ B0 年（0～99） 月（1～12） （例）95年 10月 H9510	S（发生出错）
SD2			• 将SD0数据更新的日、时以2位BCD码的形式存储。 B15 ～ B8 B7 ～ B0 日（1～31） 时（0～23） （例）25日10时 H2510	
SD3			• 将SD0数据更新的分、秒以2位BCD码的形式存储。 B15 ～ B8 B7 ～ B0 分（0～59） 秒（0～59） （例）35分48秒 H3548	
SD4	出错信息区分	出错信息区分代码	• 存储用于判别相同信息（SD5～SD15）、个别信息（SD16～SD26）中各自存储的是什么出错信息的区分编码。 B15 ～ B8 B7 ～ B0 各别信息区分代码 公共信息区分代码 • 相同信息区分代码中存入下列代码： 0：无 1：模块号/机号/基板号 2：文件名/驱动器名 3：时间（设置值） 4：程序出错处 • 个别信息区分代码中存入下列代码： 0：无 1：（空） 2：文件名/驱动器名 3：时间（实测值） 4：程序出错处 5：参数号 6：信号报警器F号	S（发生出错）

特殊寄存器一览（续）

<table>
<tr><th>编号</th><th>名称</th><th>内容</th><th>详细内容</th><th>设置侧
（设置时期）</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>通用出错信息</td><td>通用出错信息</td><td>
• 存储与出错代码（SD0）相对应的相同信息。
• 所存储的信息有以下4种类型。
① 模块号
<table>
<tr><th>代号</th><th>内容</th></tr>
<tr><td>SD5</td><td>插槽号 / 基板号</td></tr>
<tr><td>SD6</td><td>I/O号</td></tr>
<tr><td>SD7</td><td rowspan="9">（空）</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
SD6（输入/输出号）中存入0FFFFH时，有时输入/输出分配参数中可能会因为输入/输出号的重复等而无法确定输入/输出 号，此时，利用SD5确定异常部位。
② 文件名/驱动器名
<table>
<tr><th>代号</th><th colspan="2">内容</th></tr>
<tr><td>SD5</td><td colspan="2">驱动器</td></tr>
<tr><td>SD6</td><td colspan="2" rowspan="4">文件名
ASC 代码：8 字符）</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td><td>扩展符 *1</td><td>2EH(.)</td></tr>
<tr><td>SD11</td><td colspan="2">（ASCⅡ代码：3 字符）</td></tr>
<tr><td>SD12</td><td colspan="2" rowspan="4">（空）</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
（例）文件名＝
MAIN.QPG
<table>
<tr><th>B15～B8</th><th>B7～B0</th></tr>
<tr><td>41H(A)</td><td>41H(M)</td></tr>
<tr><td>4EH(N)</td><td>49H(I)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>20H(SP)</td><td>20H(SP)</td></tr>
<tr><td>51H(Q)</td><td>2DH(.)</td></tr>
<tr><td>47H(G)</td><td>50H(P)</td></tr>
</table>
③ 时间（设置值）
<table>
<tr><th>代号</th><th>内容</th></tr>
<tr><td>SD5</td><td>时间：1 s 单位 0 999 s</td></tr>
<tr><td>SD6</td><td>时间：1ms 单位 0 65535ms</td></tr>
<tr><td>SD7</td><td rowspan="9">（空）</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>
④ 程序出错处
<table>
<tr><th>代号</th><th colspan="2">内容</th></tr>
<tr><td>SD5</td><td colspan="2" rowspan="4">文件名
ASC 代码：8字符）</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td><td>扩展符 *1</td><td>2EH(.)</td></tr>
<tr><td>SD10</td><td colspan="2">ASC 代码：3字符）</td></tr>
<tr><td>SD11</td><td colspan="2">模式 *2</td></tr>
<tr><td>SD12</td><td colspan="2">块号</td></tr>
<tr><td>SD13</td><td colspan="2">步号 / 转移号</td></tr>
<tr><td>SD14</td><td colspan="2">顺控步号 L</td></tr>
<tr><td>SD15</td><td colspan="2">顺控步号 H</td></tr>
</table>
*2: 模式数据固定为0。
</td><td>S（发生出错）</td></tr>
</table>

*1：扩展名请参照下页的备注。

特殊寄存器一览（续）

编号	名称	内容	详细内容	设置侧 （设置时期）
SD16 SD17 SD18 SD19 SD20 SD21 SD22 SD23 SD24 SD25 SD26	相同出错信息	相同出错信息	• 存储与出错代码（SD0）相对应的各别信息。 • 所存储的信息有以下6种类型。 （见下方①～⑥）	S（发生出错）

① 文件名/驱动器名

代号	内容	
SD16	驱动器	
SD17	文件名 ASC 代码：8 字符）	
SD18		
SD19		
SD20		
SD21	扩展符 *1	2EH(.)
SD22	（ASC II 代码：3 字符）	
SD23	（空）	
SD24		
SD25		
SD26		

（例）文件名＝MAIN.QPG

B15～B8	B7～B0
41H(A)	41H(M)
4EH(N)	49H(I)
20H(SP)	20H(SP)
20H(SP)	20H(SP)
51H(Q)	2DH(.)
47H(G)	50H(P)

② 时间（实测值）

代号	内容
SD16	时间：1 s 单位 0 999 s
SD17	时间：1ms 单位 0 65535ms
SD18	（空）
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

③ 程序出错处

代号	内容	
SD16	文件名 ASC 代码：8字符）	
SD17		
SD18		
SD19		
SD20	扩展符 *1	2EH(.)
SD21	ASC 代码：3字符）	
SD22	模式 *2	
SD23	块号	
SD24	步号 / 转移号	
SD25	顺控步号 L	
SD26	顺控步号 H	

*2 ：模式数据固定为0。

④ 参数号

代号	内容
SD16	参数号 * 2
SD17	（空）
SD18	
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

⑤ 信号报警器号

代号	内容
SD16	No.
SD17	（空）
SD18	
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

⑥ 智能功能模块参数出错

代号	内容
SD16	参数号 * 3
SD17	智能功能模块的出错代码
SD18	（空）
SD19	
SD20	
SD21	
SD22	
SD23	
SD24	
SD25	
SD26	

*3：参数号的详细内容请参照所使用的CPU的用户手册。

备　注

*1：扩展名如下表所示。

SD10	SD11		扩展名	文件的类别
高位8位	低位8位	高位8位		
51H	50H	41H	QPA	参数
51H	50H	47H	QPG	顺控程序
51H	43H	44H	QCD	软元件注释
51H	44H	52H	QDR	文件寄存器

特殊寄存器一览（续）

编号	名称	内容	详细内容	设置侧（设置时期）
SD50	出错解除	解除出错出错号	• 存储出错解除的出错号。	U
SD51	电池电压低锁存	发生电池电压低的对象的位模式	• 发生电池低下的情况下，相对应的位全部接通。 • 此后，即使电源电压恢复正常，仍保持接通状态。 B0 0<--> → CPU出错	S（发生出错）
SD52	电池低下	发生电池电压低的对象的位模式	• 和上述SD5同样构成。 • 此后，如果电池电压恢复正常，就切断。	S（发生出错）
SD53	检出AC/DC DOWN	AC/DC DOWN次数	• CPU模块在运算中每次输入电压低于额定电压的85%（AC电源）/65%（DC电源）时＋1，其值以BIN码存储。	S（发生出错）
SD60	保险丝断路号	保险丝断路模块号	• 存储发生保险丝断路的模块的最新输入/输出号。	S（发生出错）
SD61	输入输出模块核对出错号	输入输出模块核对出错模块号	• 存储发生输入输出模块核对出错的模块的最新输入/输出号。	S（发生出错）
SD62	信号报警器号	信号报警器号	• 存储最早检测出的信号报警器号	S（执行指令）
SD63	信号报警器个数	信号报警器个数	• 存储所检测出的信号报警器的个数。	S（执行指令）
SD64 SD65 SD66 SD67 SD68 SD69 SD70 SD71 SD72 SD73 SD74 SD75 SD76 SD77 SD78 SD79	信号报警器检出编号表	信号报警器检出编号	利用 OUT F ， SET F 接通F后，在SD64～SD79中依次注册接通的F编号。 利用 RST F 从SD64～SD79中删除切断的F编号，被删除的F编号移动到原来所存储的模块以后的数据寄存器中。信号报警器的检测出个数为16个的情况下，即使检测出第17个，也不存入SD64～SD79。	S（执行指令）
SD100	传输速度存储区域		96 ：9.6kbps， 192 ：19.2kbps， 384：38.4kbps， 576 ：57.6kbps，1152 ：115.2kbps	S（电源接通、复位解除时）
SD101	通信设置存储区域	存储串行通信设置时指定的通信设置	（见下图）	S（电源接通、复位解除时）
SD102	报文等待时间存储区域	存储串行通信设置时指定的报文等待时间	0：无等待时间 1～FH：等待时间（单位：10 ms） 默认值为0	S（电源接通、复位解除时）

SD64～SD79 示例：

		SET F50	SET F25	SET F99	RET F25	SET F15	SET F70	SET F65	SET F38	SET F110	SET F151	SET F210	LEDR	
SD62	0	50	50	50	50	50	50	50	50	50	50	50	99	（检测号）
SD63	0	1	2	3	2	3	4	5	6	7	8	9	8	（检测个数）
SD64	0	50	50	50	50	50	50	50	50	50	50	50	99	（检测号）
SD65	0	0	25	25	99	99	99	99	99	99	99	99	15	
SD66	0	0	0	99	0	15	15	15	15	15	15	15	70	
SD67	0	0	0	0	0	0	70	70	70	70	70	70	65	
SD68	0	0	0	0	0	0	0	65	65	65	65	65	38	
SD69	0	0	0	0	0	0	0	0	38	38	38	38	110	
SD70	0	0	0	0	0	0	0	0	0	110	110	110	151	
SD71	0	0	0	0	0	0	0	0	0	0	151	151	210	
SD72	0	0	0	0	0	0	0	0	0	0	0	210	0	
SD73	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD74	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD75	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD76	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD77	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD78	0	0	0	0	0	0	0	0	0	0	0	0	0	
SD79	0	0	0	0	0	0	0	0	0	0	0	0	0	

SD101 位构成：

F　6 5 4 3 2 1 0

- 位5：RUN 中写入设置　0：禁止　1：允许
- 位4：有无和校验　0：无　1：有

特殊寄存器一览（续）

编号	名称	内容	详细内容	设置侧 （设置时期）
SD110	数据发送结果 存储区域	存储使用串行通信功能时的数据发送结果	存储采用串行通信功能发送数据时的出错代码。	S（发生出错）
SD111	数据接收结果 存储区域	存储使用串行通信功能时的数据接收结果	存储数据接收时的出错代码。	S（发生出错）
SD130 SD131 SD132 SD133 SD134 SD135 SD136 SD137	保险丝断路 模块	保险丝断路模块的16点单位的位模式 0：无保险丝断路 1：有保险丝断路	• 处于保险丝断路状态的输出模块编号（16点单位）以位模式的形式进入。（参数设置时位所设置的编号） • 远程站的输出模块的保险丝断路也能检出。 B15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD130: 0 0 0 1(YC0) 0 0 0 1(Y80) 0 0 0 0 0 0 0 0 SD131: 1(Y1F0) 0 0 0 0 1(Y1A) 0 0 0 0 0 0 0 0 0 0 SD137: 0 0 0 0 1(Y7B0) 0 0 0 0 0 0 0 1(Y730) 0 0 0 表示保险丝融断状态 • 即使恢复正常也不清零。 出错解除后被清零。	S（发生出错）
SD150 SD151 SD152 SD153 SD154 SD155 SD156 SD157	输入输出模块 核对出错	核对出错模块的16点单位的位模式 0：无输入输出核对出错 1：有输入输出核对出错	• 检测出和电源接通时注册的输入输出模块信息不同的输入输出模块时，存入该输入输出模块编号（16点单位）。 （如果是用常数设置的，就是设置的输入输出模块编号） B15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD150: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1(XY0) SD151: 0 0 0 0 0 0 1(XY190) 0 0 0 0 0 0 0 0 0 SD157: 0 0 0 0 1(XY7E0) 0 0 0 0 0 0 0 0 0 0 0 表示输入输出模块核对出错 • 即使恢复正常也不清零。 出错解除后被清零。	S（发生出错）

特殊寄存器一览

(2) 系统信息

编号	名称	内容	详细内容	设置侧 （设置时期）
SD200	开关状态	CPU开关状态	• 采用以下格式存储CPU的开关状态。 B15 B8 B7 B4 B3 B0 空 ② ① ①：CPU开关状态 0：运行 1：停止 ②：存储卡开关 常开	S（每次结束）
SD201	LED状态	CPU LED状态	• 采用以下的位模式存储CPU的LED处于以下之中的何种状态。 • 0表示熄灭；1表示点亮；2表示闪烁。 B15 B4 B3 B0 空 ② ① ①：运行 ②：ERROR	S（状态变化）
SD203	CPU动作状态	CPU动作状态	• 如下图，存储CPU的动作状态。 B15 B12 B11 B8 B7 B4 B3 B0 ② ① ①：CPU动作状态 0：运行 1：空 2：停止 3：暂停 ②：停止/暂停原因 0：开关 1：远程触点 2：来自GX Developer/串行通信模块等的远程操作 3：程序中的指令 4：出错 注）先者优先	S（每次结束）
SD210	时钟数据	时钟数据 （公历，月）	• 如下图，以BCD码形式在SD210中存储年（公历末尾2位）和月。 B15 …… B12 B11 …… B8 B7 …… B4 B3 …… B0 年 月 （例）93年7月 H9307	S/U（请求时）
SD211	时钟数据	时钟数据 （日，时）	• 如下图，以BCD码形式在SD211中存储日和时。 B15 …… B12 B11 …… B8 B7 …… B4 B3 …… B0 日 时 例）31日、10时 H3110	
SD212	时钟数据	时钟数据 （分，秒）	• 如下图，以BCD码形式在SD211中存储分和秒。 B15 …… B12 B11 …… B8 B7 …… B4 B3 …… B0 分 秒 例）35分，48秒 H3548	

特殊寄存器一览（续）

编号	名称	内容	详细内容	设置侧 （设置时期）
SD213	时钟数据	时钟数据 （公历高位、星期几）	• 如下图，以BCD码形式在SD213中存储年（公历的高位2位）和星期几。 B15 ⋯⋯ B12 B11 ⋯⋯ B8 B7 ⋯⋯ B4 B3 ⋯⋯ B0　例）星期五 H0005 西历上位 0　99 星期：0 星期日 / 1 星期一 / 2 星期二 / 3 星期三 / 4 星期四 / 5 星期五 / 6 星期六	S/U（请求时）
SD220 SD221 SD222 SD223 SD224 SD225 SD226 SD227	显示器数据	显示器数据	• 存储发生出错（也包括信号报警器接通）时的信息（16个ASCII字符）。 B15 ～ B8　B7 ～ B0 SD220：从右起第 15 个字符 / 从右起第 16 个字符 SD221：从右起第 13 个字符 / 从右起第 14 个字符 SD222：从右起第 11 个字符 / 从右起第 12 个字符 SD223：从右起第 9 个字符 / 从右起第 10 个字符 SD224：从右起第 7 个字符 / 从右起第 8 个字符 SD225：从右起第 5 个字符 / 从右起第 6 个字符 SD226：从右起第 3 个字符 / 从右起第 4 个字符 SD226：从右起第 1 个字符 / 从右起第 2 个字符 • PRG、CHK时的显示器数据不存储。	S（变化时）
SD240	基本模式	0：自动模式 1：详细模式	存储基本模式。	S（开始时）
SD241	扩展级数	0：只有主板 1～4：扩展级数	• 存储所装载的扩展基板的最大级数。	S（开始时）
SD242	A/Q基板判别	基板类型的判别 0：未安装 1：安装Q**B	B4　B2　B1　B0 0 固定 ～ B0 → 主基板 B1 → 扩展 1 级 B2 → 扩展2 级 ～　未安装时为 0 固定 B4 → 扩展 4 级	S（开始时）
SD243 SD244	基板插槽数	基板插槽数	B15 B12　B11 B8　B7 B4　B3 B0 SM243：扩展 3 / 扩展 3 / 扩展 3 / 主基板 SM244：0 固定 / 0 固定 / 0 固定 / 扩展 4 • 在上述各区域中存储所装载的基板的插槽数。	S（开始时）
SD250	装载的最大输入/输出	装载的最大输入/输出号	• 存储所装载的模块的最终输入/输出编号＋1的高位2位。	S（开始时）
SD254	MELSECNET/H信息	安装块数	• 表示所装载的MELSECNET/H的块数。	S（开始时）
SD255		输入/输出号	• 所装载的MELSECNET/H的输入/输出号	
SD256		网络号	• 所装载的MELSECNET/H的网络号	
SD257		组号	• 所装载的MELSECNET/H的组号	
SD258		站号	• 所装载的MELSECNET/H的站号	

特殊寄存器一览（续）

编号	名称	内容	详细内容	设置侧 （设置时期）
SD290	软元件分配 （和参数内容相同）	X分配点数	• 存储当前已设置的软元件X的点数。	S（初期）
SD291		Y分配点数	• 存储当前已设置的软元件Y的点数。	
SD292		M分配点数	• 存储当前已设置的软元件M的点数。	
SD293		L分配点数	• 存储当前已设置的软元件L的点数。	
SD294		B分配点数	• 存储当前已设置的软元件B的点数。	
SD295		F分配点数	• 存储当前已设置的软元件F的点数。	
SD296		SB分配点数	• 存储当前已设置的软元件SB的点数。	
SD297		V分配点数	• 存储当前已设置的软元件V的点数。	
SD298		S分配点数	• 存储当前已设置的软元件S的点数。	
SD299		T分配点数	• 存储当前已设置的软元件T的点数。	
SD300		ST分配点数	• 存储当前已设置的软元件ST的点数。	
SD301		C分配点数	• 存储当前已设置的软元件C的点数。	
SD302		D分配点数	• 存储当前已设置的软元件D的点数。	
SD303		W分配点数	• 存储当前已设置的软元件W的点数。	
SD304		SW分配点数	• 存储当前已设置的软元件SW的点数。	
SD315	通信处理确保时间	通信处理确保时间	确保指定的GX Developer等的通信处理时间。 所指定的值越大，和其他软元件（GX Developer、串行通信模块等）的通信响应速度越快。 设置范围：1～100 ms 超出上述范围设置的情况下，作为未设置处理。 但是，扫描时间将延长指定时间。	U（结束处理）
SD340	Ethernet信息	安装块数	• 表示所装载的Ethernet的块数。	S（初期）
SD341		I/O No.	• 表示所装载的Ethernet的输入/输出号	
SD342		网络No.	• 表示所装载的Ethernet的网络号	
SD343		组No.	• 表示所装载的Ethernet的组号	
SD344		站号	• 表示所装载的Ethernet的站号	

（3）系统时钟/计数器

编号	名称	内容	详细内容	设置侧 （设置时期）
SD412	1秒计数器	1秒单位的计数值	• 可编程控制器CPU 运行后，每秒＋1。 • 计数器反复0→32767→—32768→0。	S（状态变化）
SD414	2n秒计数器	2n秒时钟的单位	• 存储2n秒时钟的n。（默认值为30） • 可以在1～32767的范围内设置。	U
SD420	扫描计数器	每次扫描的计数值	• 可编程控制器CPU 运行后，每秒＋1。 • 计数器反复0→32767→—32768→0。	S（每次结束）

（4）扫描信息

编号	名称	内容	详细内容	设置侧（设置时期）
SD520	当前扫描时间	当前扫描时间（1ms单位）	• 存储当前的扫描时间。（1 ms单位） • 0～65535的范围	S（每次结束）
SD521	当前扫描时间	当前扫描时间（100μs单位）	• 存储当前的扫描时间。（100 μs单位） • 000～900的范围 （例）当前的扫描时间、时间为23.6 ms的情况下，存储方法如下： D520＝23 D521＝600	S（每次结束）
SD524	最小扫描时间	最小扫描时间（1ms单位）	• 存储扫描时间的最小值。（1 ms单位） • 0～65535的范围	S（每次结束）
SD525	最小扫描时间	最小扫描时间（100μs单位）	• 存储扫描时间的最小值。（100 μs单位） • 000～900的范围	S（每次结束）
SD526	最大扫描时间	最大扫描时间（1ms单位）	• 除第1次扫描外，存储扫描时间的最大值。（1 ms单位） • 0～65535的范围	S（每次结束）
SD527	最大扫描时间	最大扫描时间（100μs单位）	• 除第1次扫描外，存储扫描时间的最大值。（100 μs单位） • 000～900的范围	S（每次结束）
SD540	结束处理时间	结束处理时间（1ms单位）	• 扫描程序结束后，存储下次扫描开始之前的时间。（1 ms单位） • 0～65535的范围	S（每次结束）
SD541	结束处理时间	结束处理时间（100μs单位）	• 扫描程序结束后，存储下次扫描开始之前的时间。（100 μs单位） • 000～900的范围	S（每次结束）
SD542	恒定扫描等待时间	恒定扫描等待时间（1ms单位）	• 存储恒定扫描设置时的等待时间。（1 ms单位） • 0～65535的范围	S（每次结束）
SD543	恒定扫描等待时间	恒定扫描等待时间（100μs单位）	• 存储恒定扫描设置时的等待时间。（100 μs单位） • 000～900的范围	S（每次结束）
SD548	扫描程序执行时间	扫描程序执行时间（1ms单位）	• 存储1次扫描期间的扫描程序执行时间。（1 ms单位） • 0～65535的范围 • 每次扫描时存储	S（每次结束）
SD549	扫描程序执行时间	扫描程序执行时间（100μs单位）	• 存储1次扫描期间的扫描程序执行时间。（100 μs单位） • 000～900的范围 • 每次扫描时存储	S（每次结束）

特殊寄存器一览

(5) 存储卡

编号	名称	内容	详细内容	设置侧（设置时期）
SD620	存储卡B类别	存储卡B的类别	• 表示所安装的存储卡B的类别。 B15～B8: 0<----------->0; B7～B4: 驱动器4（标准ROM）因为内置标准ROM 3 固定; B3～B0: 驱动器3（标准RAM）0：无 1：有	S（开始时）
SD622	驱动器3(标准RAM)容量	驱动器3的容量	• 以1 k字节单位存储驱动器3的容量。	S（开始时）
SD623	驱动器4（标准ROM）容量	驱动器4的容量	• 以1 k字节单位存储驱动器4的容量。	S（开始时）
SD624	驱动器3 使用状况	驱动器3的使用状况	• 以位模式存储驱动器3的使用状况。 B15～B5: 0; B4～B0: 0 0 0 0 0 B0: 文件寄存器(R) 1:在使用 0:未使用	S（状态变化）
SD640	文件寄存器 驱动器	驱动器号	• 存储文件寄存器所使用的驱动器号。	S（状态变化）*
SD641 SD642 SD643 SD644 SD645 SD646	文件寄存器 文件名	文件寄存器 文件名	• 以ASCII码形式存储文件寄存器的文件名。 (B15～B8 / B7～B0) SD641: 第 2 个字符 / 第 1 个字符 SD642: 第 4 个字符 / 第 3 个字符 SD643: 第 6 个字符 / 第 5 个字符 SD644: 第 8 个字符 / 第 7 个字符 SD645: 扩展符第 1 个字符 / 2EH(.) SD646: 扩展符第 3 个字符 / 扩展符第 2 个字符	S（状态变化）*
SD647	文件寄存器容量	文件寄存器的容量	• 以1 k字单位存储当前所选择的文件寄存器的数据容量。	S（状态变化）*
SD648	文件寄存器 块号	文件寄存器的 块号	• 存储当前所选择的文件寄存器的块号。	S（状态变化）*

*：参数执行后从停止→运行或RSET指令执行时设置数据。

特殊寄存器一览

(6) 指令相关

编号	名称	内容	详细内容	设置侧 （设置时期）
SD715 SD716 SD717	IMASK指令 屏蔽模式	屏蔽模式	• 利用IMASK指令屏蔽的屏蔽模式按如下方法存储。 B15 B11 B0 SD715: I15 ～ I1 I0 SD716: I31 ～ I17 I16 SD717: I47 ～ I33 I32	S（执行时）
SD718 SD719	累加器	累加器	• A系列的程序中的累加器变换用。	S/U
SD781 ～ SD785	IMASK指令 屏蔽模式	屏蔽模式	• 利用IMASK指令屏蔽的屏蔽模式按如下方法存储。 B15 B1 B0 SD781: I63 ～ I49 I48 SD782: I79 ～ I65 I64 ～ SD785: I127 ～ I113 I112	S（执行时）

附录3 中断指针编号和中断原因一览

中断号	中断原因		优先顺序
I0	由QI60引起的中断	第1点	1
I1		第2点	2
I2		第3点	3
I3		第4点	4
I4		第5点	5
I5		第6点	6
I6		第7点	7
I7		第8点	8
I8		第9点	9
I9		第10点	10
I10		第11点	11
I11		第12点	12
I12		第13点	13
I13		第14点	14
I14		第15点	15
I15		第16点	16
I16～I27	不可使用	—	—
I28	由内部定时器引起的中断＊1	100ms	20
I29		40ms	19
I30		20ms	18
I31		10ms	17
I32～I127	不可使用	—	—

备　　注

＊1：内部定时器的时限为默认值。
　　PC参数的PC系统设置时可以在2 ms～1000 ms的范围内以1 ms单位变化。

索引

索引 - 1

索引 - 2

索引 - 3

备忘录

索引 – 4

质保

使用之前请确认下述产品质保的细节：

1．免费质保期限和免费质保范围

如果是在质保期内使用本产品时发现因[三菱电机]的责任而导致的异常或缺陷（下文一并简称为“故障”），则该产品应该由经销商或[三菱电机]维修公司免费维修。注意如果需要派员到海外，孤立的岛屿或者偏远地方，则要收取派遣技术人员的费用。

【免费质保期】

本产品的免费质保期为一年，自购买或货到目的地的日期起算。

注意从制造并运出[三菱电机]开始，最长分销时间不得超过6个月，从制造之日开始的最长免费质保期不得超过18个月。

经过修理的产品的免费质保期不得超过修理以前的免费质保期。

【免费质保范围】

（1） 范围被限制在按照使用手册、用户手册和产品上的警示标贴上规定的使用状态、使用方法和使用环境正常使用的条件下。

（2） 即便在免费质保期内，下列情况下修理要收费。

① 因不合理存储或搬运、用户的大意或疏忽而导致的故障。因用户的硬件或软件设计而导致的故障。

② 因用户未经批准对该产品进行改造而引起的故障。

③ 把[三菱电机]产品装配在用户设备中时，如果用户设备根据法律安全条款或工业标准要求配备必需的功能和结构，故障本来可以避免时。

④ 如果正确采用或更换了用户手册中指定的耗材（电池、背光灯、保险丝等）故障本来可以避免时。

⑤ 因火灾、不正常电压等外部因素和因地震、雷电、大风和水灾等引起的不可抗力引发的故障。

⑥ 按照产品从[三菱电机]出厂时的科技水平不能预测的原因而导致的故障。

⑦ 任何不是因[三菱电机]或用户认为非本公司责任导致的故障。

2．停止产品生产以后的有偿修理期限

（1） [三菱电机]在本产品停止生产后的7年内受理对该产品的有偿修理。停止生产的消息将以[三菱电机]技术公告等方式予以通知。。

（2） 生产停止以后，不再提供产品（包括修理用零部件）。

3．海外服务

在海外，修理由[三菱电机]在当地的海外FA中心受理。请注意各个FA中心的修理条件可能会有所不同。

4．机会损失和间接损失不在质保责任范围内

不论是否在免费保质期内，[三菱电机]对任何不是[三菱电机]的责任的原因而引起的损失、因[三菱电机]产品故障而导致的客户的机会损失利润损失、违反[三菱电机]要求的特殊原因而引起的损失或间接损失、事故赔偿、及非[三菱电机]的其它产品的损坏和赔偿等不承担责任。

5．产品规格的改变

目录、手册或技术文档中的规格的改变不事先通知。

6．产品的适用性

（1）在使用[三菱电机]MELSEC通用可编程逻辑控制器时，应该符合下列条件：即便可编程控制器出现问题或故障也不会导致重大事故，并且应在设备外部系统地配备能就应付任何问题或故障的备用设施和失效保险功能。

（2）三菱通用可编程序控制器是一般工业用途为对象设计和制造的。因此，可编程序控制器的应用不包括那些会影响公众利益的应用如核电厂和其他由独立供电公司经营的电厂以及需要特殊质量控制系统的应用如铁路公司或用于国防目的的应用。

请注意即便是这些应用，假如用户同意该应用受限制并且不需要特别质量的话，仍然可以作这类应用。

在用于航空、医学、铁路、焚烧和燃料设备，传送人的设备，娱乐和休闲设施和安全设施等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时，请与三菱公司联系并互相交换必要的规格书等资料。

以　　上

Q 系列精简模式 CPU

用户手册（功能解说、编程基础篇）

型号	SQCPU(Q)-U-KI-C
	SH(NA)-080331C-A



HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212 , JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN